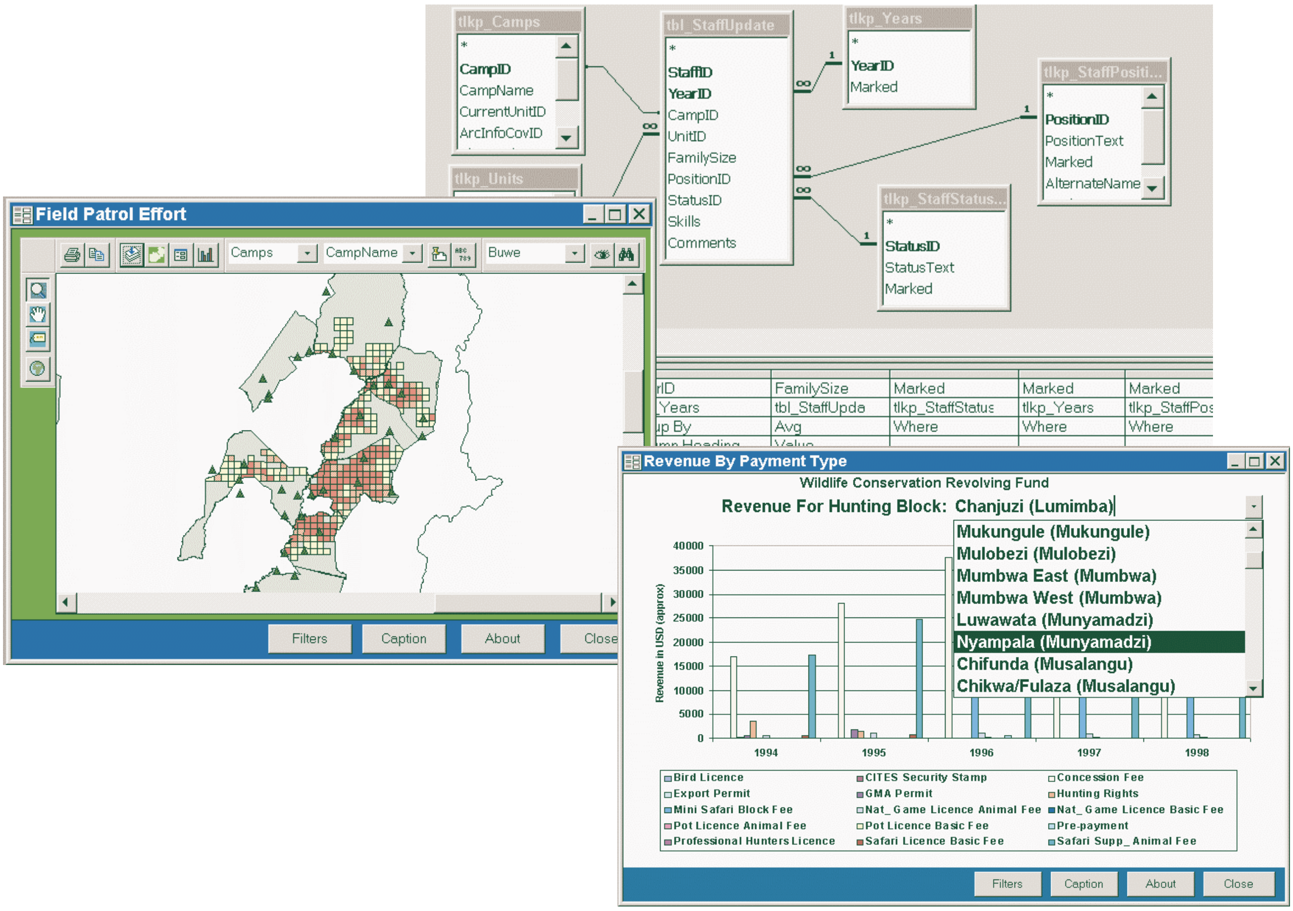
ADMADE DATA MANAGER

Users Guide



National Parks and Wildlife Services
Nyamaluma Institute for Community Based Resource Management
P.O. Box 82
Mfuwe, Zambia
May, 1999

Table of Contents

Chapter 1: About The ADMADE Data Manager

The ADMADE Data Manager (ADM) is a MS Access application used to manage information for the ADMADE wildlife conservation program in Zambia. A database application such as ADM can be defined as a group of database objects (tables, data entry forms, menu system, reports, queries) that are designed to work together and use the built-in capabilities of MS Access to manage and analyze information.

ADM was introduced in early 1999, but is built upon ADMADEs former information system which used a combination of Lotus spreadsheets, dBase for DOS, and ArcView GIS for data entry and summaries. Almost all of ADMADE's data collected from 1991-1998 have been converted for use with ADM. Some of the main characteristics of ADM include:

- ADM is designed to be both a working day-to-day data management system, as well as provide powerful analytical tools to study the ADMADE program's performance in wildlife management and community development. It has features such as automatic data entry logging as well as a variety of tools for presentation of data.
- ADM is a relatively user-friendly system, having a menu system that makes it easy to enter or retrieve information. The Filter Manager makes selecting records for analysis flexible and simple.
- ADM is a relational database system, meaning that different types of data (e.g., staff records, demography, field patrol data) are saved in separate tables, However related information can be easily linked back together for analysis using primary key fields.
- ADM makes full use of virtually all of Access's built-in features, including enforcing data integrity between related tables, replication, and using the Visual Basic for Applications (VBA) programming language for program navigation and facilitating extensibility.
- ADM is designed to grow and expand as new needs arise. The Menu System makes it easy to integrate new queries, graphs, and reports and integrate them into the existing interface.
- ADM is well-equipped to display spatial information (i.e., interactive maps and charting), but is not designed to replace the functions of a full featured GIS package such as ArcView.

ADMADE's information and monitoring system is the result of the efforts of many people over many years, including Nyamaluma staff, its long-term technical advisor, and numerous short term volunteers and consultants. The ADM Access interface, its menu and filter system, and most of this users guide were developed by Andy Lyons (alyons@nersp.nerdc.ufl.edu), a masters student from the University of Florida who conducted research with ADMADE 1998/99.

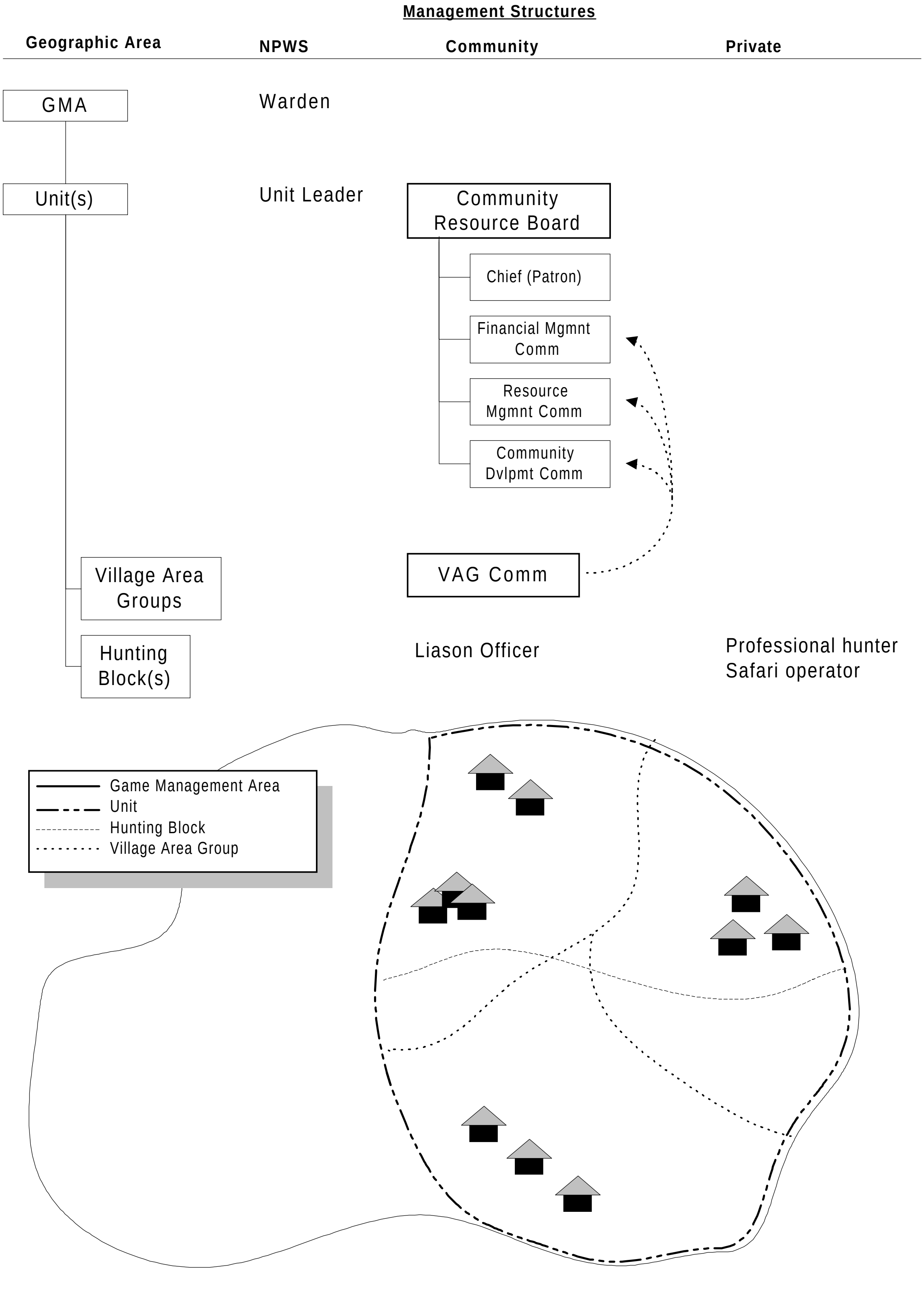
About ADMADE

ADMADE is a program (not a separate institution) of the Zambia Wildlife Authority (formerly known as the National Parks and Wildlife Services). The ultimate objective of ADMADE is to conserve wildlife in Game Management Areas through a partnership of the private safari hunting industry, government, and local communities. Operationally, a portion of the revenue from safari hunters is returned to the local communities through the Wildlife Conservation Revolving Fund (WCRF), who use the money for community development projects and law enforcement.

The primary level of structure in ADMADE is the Unit, which in most cases also corresponds to a single Game Management Area. As illustrated below, the ADMADE organizational design calls for a three-tiered structure of community based organizations within a Unit. Village Action Groups

(VAG) are comprised of elected representatives from a cluster of villages. Leadership of the VAG in turn belongs to Unit wide committees for financial management, community development, and resource management. These three management committees then make recommendations to the Community Resource Board, another elected body which has final say over how financial resources are used.

ADMADE Organization



ADMADE is operational in about 22 of the 39 GMAs in Zambia. Training and oversight of the program is provided by the Nyamaluma Institute for Community Based Resource Management, a ZWA/NPWS training facility near Mfuwe in Eastern Province. The success of ADMADE in implementing its activities has varied significantly across GMAs. It is operating most successfully in about 12 GMAs, primarily in the Luangwa Valley. USAID, Wildlife Conservation Society (Bronx Zoo), and World Wildlife Fund are the donors and NGOs have been the primary donors and NGOs active with ADMADE.

For more information about ADMADE, including a listing of published literature, see the ADMADE web site http://www.admade.org.zm.

Data Sets

ADM stores a wide range of data, from demography to the sizes of hunting trophies. The primary data sets which are maintained in the ADMADE Data Manager include:

- **Field Patrols** – ADMADE Village Scouts collect data while on anti-poaching patrols. They record information such as dates and location of the patrol, supplies taken, location of poacher camps, carcasses of animals, and sightings of live animals

- **Safari Hunting Results** – Whenever a safari client goes on a hunt, an ADMADE Village scout accompanies the party to record information such as location of the kill, search time, and trophy size. This information is used to help evaluate the sustainability of safari hunting and set quotas.

- **WCRF License Sales and Prices** – Each time a client or professional hunter purchases hunting licenses for a safari hunt, a record is made in the Wildlife Conservation Revolving Fund database. This database is housed at the licensing office in Chilanga (Lusaka), and a portion of the money is returned to ADMADE Units for resource management and community development.

- **Staff Histories** – each year the staff list for each ADMADE Unit is updated. Information which is recorded on individual employees includes their position, camp, rank, and highest level of education

- **Quotas** – hunting quotas specify the number of animals which can be legally hunted. Quotas are set by the NPWS and are updated once a year for each hunting block

- **Demography** – For some Village Area Groups, household demography data is available. This information is collected in door to door surveys by the CBDs who contracted to do the census.

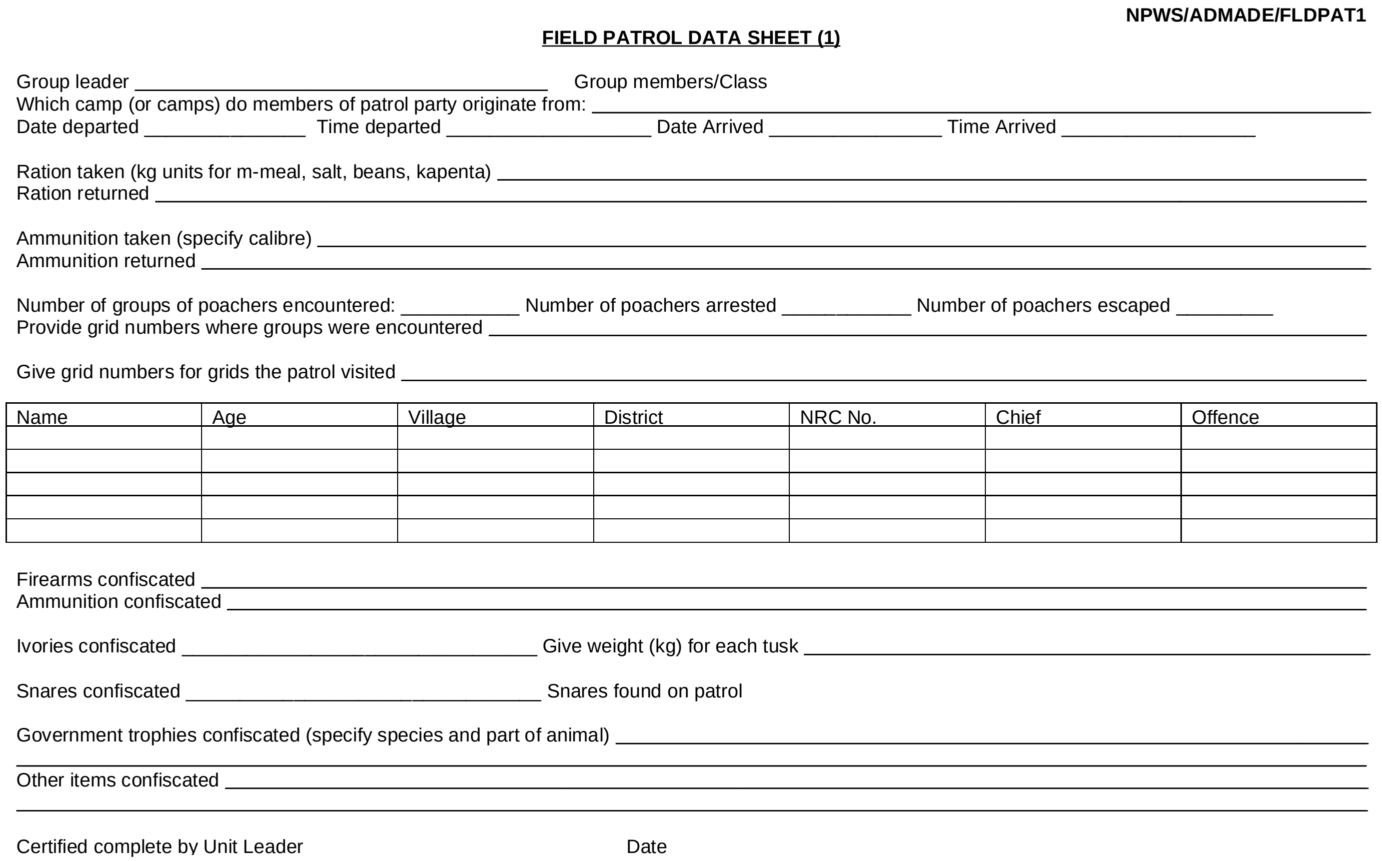
- **Species Biology** – This table contains basic biological parameters for many of the game species, including average body mass, minimum measurements for trophy specimens, gestation interval, water dependency, etc. This data can be used for biometric analyses and such.

- **Documents** – several field reports, land use plans, correspondence, and assorted other documents are available for many of the ADMADE Units.

This different datasets are saved in various tables in ADM. To learn what these tables contain and how they work together, view the table definition and check the description property for the fields and the table itself.

FLDPAT1

The FLDPAT1 dataform is supposed to be filled for each Field Patrols. In practice, it is not known exactly what percentage of field patrols are actually recorded, but we can safely assume that there is a good deal of variation. FLDPAT1 has not changed significantly since it was first introduced in 1991/92, however much of the data was not entered into the database until 1999. In particular, prior to 1999 grid locations were not recorded for each patrol. Instead a summary of grid locations for the entire year was made. Also poacher information was not entered until 1999 except for the total number of poachers caught.



NPWS/ADMADE/FLDPAT1

FIELD PATROL DATA SHEET (1)

Group leader ____________________ Group members/Class
Which camp (or camps) do members of patrol party originate from: ____________________
Date departed __________ Time departed __________ Date Arrived __________ Time Arrived __________

Ration taken (kg units for m-meal, salt, beans, kapenta) ____________________
Ration returned ____________________

Ammunition taken (specify calibre) ____________________
Ammunition returned ____________________

Number of groups of poachers encountered: __________ Number of poachers arrested __________ Number of poachers escaped __________
Provide grid numbers where groups were encountered ____________________

Give grid numbers for grids the patrol visited ____________________

Name	Age	Village	District	NRC No.	Chief	Offence

Firearms confiscated ____________________
Ammunition confiscated ____________________

Ivories confiscated ____________________ Give weight (kg) for each tusk ____________________

Snares confiscated ____________________ Snares found on patrol

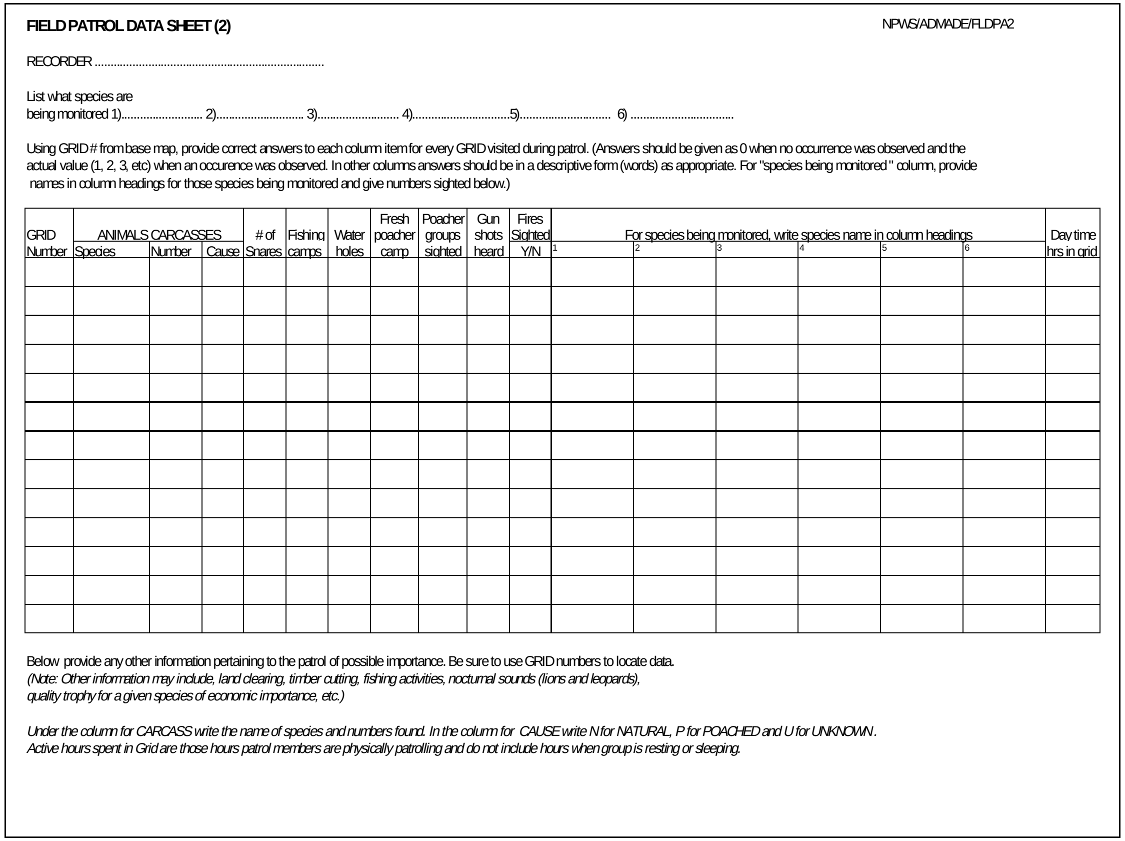
Government trophies confiscated (specify species and part of animal) ____________________
____
Other items confiscated ____________________
____

Certified complete by Unit Leader Date

FLDPAT2

A FLDPAT2 dataform is also supposed to be filled out for each field patrol, and is usually printed on the back of FLDPAT1. This form primarily records observations from field patrols. While the main purpose of field patrols is to seek and arrest poachers, scouts nevertheless generally cover a large area and are able to record their observations.

FLDPAT2 was introduced around the same time as FLDPAT1, however the first version prompted observations in a comments section instead of a structured table. In 1997, the form was modified to its present structure, however data from FLDPAT2 was not entered into the computer until 1999. In 1999, the form was again slightly modified to include the number of daylight hours in each grid, as well as the suspected cause of death for dead animals.



FIELD PATROL DATA SHEET (2)

NPWS/ADMADE/FLDPA2

RECORDER ....

List what species are
being monitored 1).......................... 2)............................ 3).......................... 4)..............................5)............................ 6) ....

Using GRID # from base map, provide correct answers to each column item for every GRID visited during patrol. (Answers should be given as 0 when no occurrence was observed and the actual value (1, 2, 3, etc) when an occurence was observed. In other columns answers should be in a descriptive form (words) as appropriate. For "species being monitored " column, provide names in column headings for those species being monitored and give numbers sighted below.)

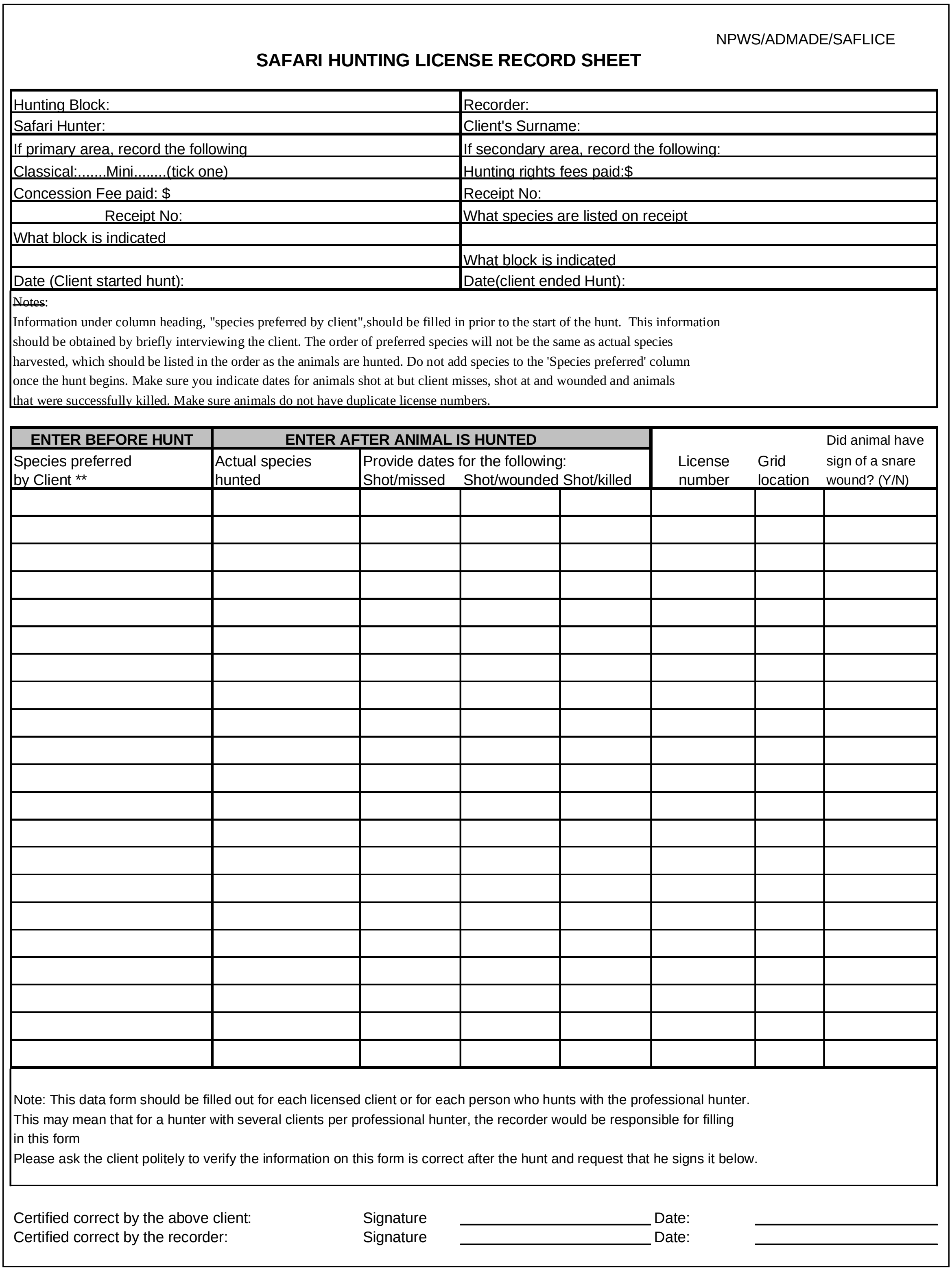
							Fresh	Poacher	Gun	Fires							
GRID	ANIMALS CARCASSES			# of	Fishing	Water	poacher	groups	shots	Sighted	For species being monitored, write species name in column headings						Day time
Number	Species	Number	Cause	Snares	camps	holes	camp	sighted	heard	Y/N	1	2	3	4	5	6	hrs in grid

Below provide any other information pertaining to the patrol of possible importance. Be sure to use GRID numbers to locate data.
(Note: Other information may include, land clearing, timber cutting, fishing activities, nocturnal sounds (lions and leopards), quality trophy for a given species of economic importance, etc.)

Under the column for CARCASS write the name of species and numbers found. In the column for CAUSE write N for NATURAL, P for POACHED and U for UNKNOWN.
Active hours spent in Grid are those hours patrol members are physically patrolling and do not include hours when group is resting or sleeping.

SAFLICE

SAFLICE is one of the main forms for safari monitoring. A village scout who accompanies safari hunters fills out one SAFLICE for each client. The list of species desired is supposed to be filled out prior to the hunt in the first column on the left, while the remaining columns are filled out in the order the animals are shot. This form is used to calculate hunting success and hunting effort.



NPWS/ADMADE/SAFLICE

SAFARI HUNTING LICENSE RECORD SHEET

Hunting Block:	Recorder:
Safari Hunter:	Client's Surname:
If primary area, record the following	If secondary area, record the following:
Classical:.......Mini........(tick one)	Hunting rights fees paid:$
Concession Fee paid: $	Receipt No:
Receipt No:	What species are listed on receipt
What block is indicated	
	What block is indicated
Date (Client started hunt):	Date(client ended Hunt):

Notes:
Information under column heading, "species preferred by client",should be filled in prior to the start of the hunt. This information should be obtained by briefly interviewing the client. The order of preferred species will not be the same as actual species harvested, which should be listed in the order as the animals are hunted. Do not add species to the 'Species preferred' column once the hunt begins. Make sure you indicate dates for animals shot at but client misses, shot at and wounded and animals that were successfully killed. Make sure animals do not have duplicate license numbers.

ENTER BEFORE HUNT	ENTER AFTER ANIMAL IS HUNTED						Did animal have
Species preferred	Actual species	Provide dates for the following:			License	Grid	sign of a snare
by Client **	hunted	Shot/missed	Shot/wounded	Shot/killed	number	location	wound? (Y/N)

Note: This data form should be filled out for each licensed client or for each person who hunts with the professional hunter.
This may mean that for a hunter with several clients per professional hunter, the recorder would be responsible for filling in this form
Please ask the client politely to verify the information on this form is correct after the hunt and request that he signs it below.

Certified correct by the above client: Signature ______________________ Date: ______________________
Certified correct by the recorder: Signature ______________________ Date: ______________________

SAFHUNT

A separate SAFHUNT dataform is supposed be filled out for each day of a safari hunt, again by the village scout accompanying the professional hunter and client. This dataform is used to cross-check information from SAFLICE, as well as note other observation on the hunt. SAFHUNT was introduced in 199x, but was not entered into the computer until 1999.

NPWS/ADMADE/SAFHUNT

SAFARI HUNTING DAILY RECORD SHEET

(Note: This form should be completed by the end of each hunting day.)

I – HUNT DESCRIPTION

Unit ______________________________ Safari operator ______________________________________
Date ______________________________ Approx hours spend hunting __________________________
Recorder __________________________ Professional hunter __________________________________
Total number of tourists (clients and observers) in hunting party ________________________________
Names of clients ____
Names of observers ____
____

Observations

1. Sightings of huntable trophies but not hunted species being monitored:
   Species _______________ Grid _______________
   Species _______________ Grid _______________
   Species _______________ Grid _______________
   Species _______________ Grid _______________
2. Snares found __________________ Grid locations ____
   Were they collected (yes/no)
3. Poacher group encountered (yes/no) ________ If yes, grid locations _________________________
4. Licensed hunters encountered (yes/no) ______ If yes, grid locations _________________________
   Were they a disturbance to client (yes/no) ______ If yes, give details ___________________________
   ____
   ____
5. Provide details and grid locations to any other human caused disturbance to the safari hunt
   ____
   ____
   ____

II – HUNTING RESULTS

License #	Person who fired gun	Species hunted	Sex	Grid location	Calibre	Successfully killed (yes/no)	Wounded (yes/no)	Hunted as trophy, bait, or both

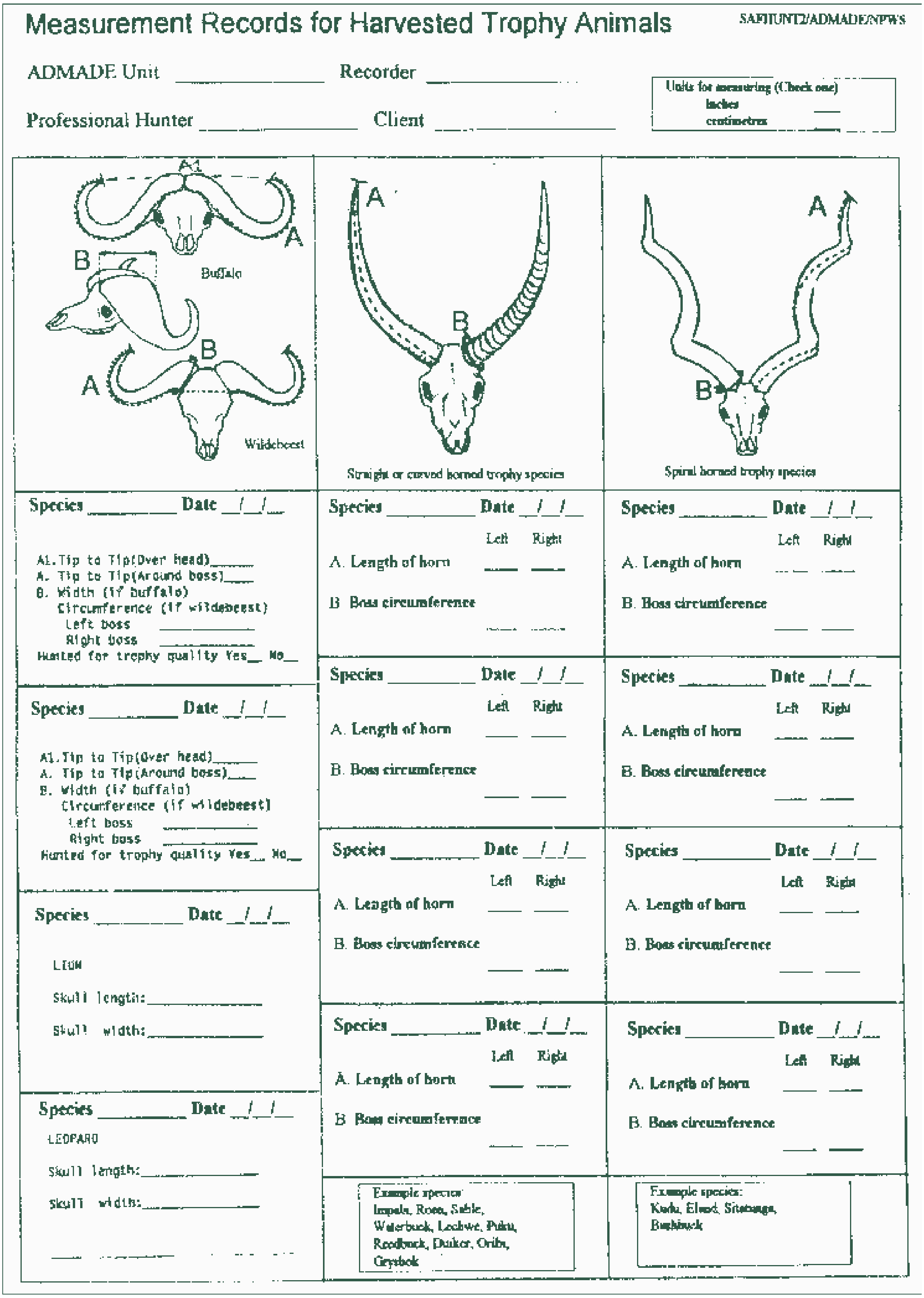
III – OBSERVATIONS AT BLINDS (lions and leopards)

Species used for bait __________________________ Grid locations of baits visited ________________
Observations

Grid #	# lions visited	# trophy lions seen	# leopards visited	Methods for securing bait (wire or chain)

TROPHY

TROPHY is the third dataform used to monitor safari hunting. It is primarily a record of trophy sizes, and has changed little since it was introduced in 1991/2. Measurement data from trophy has always been entered into the database.



Measurement Records for Harvested Trophy Animals

SAFIUNT2/ADMADE/NPWS

ADMADE Unit ______________ Recorder ______________

Professional Hunter ______________ Client ______________

Units for measuring (Check one)
Inches ___
centimetres ___

A1
A
B
Buffalo
B
A
Wildebeest
A
B
A
B

Straight or curved horned trophy species

Spiral horned trophy species

Species __________ Date __/__/__

A1. Tip to Tip(Over head)______
A. Tip to Tip(Around boss)______
B. Width (if buffalo)
Circumference (if wildebeest)
Left boss ______
Right boss ______
Hunted for trophy quality Yes__ No__

Species __________ Date __/__/__

A1. Tip to Tip(Over head)______
A. Tip to Tip(Around boss)______
B. Width (if buffalo)
Circumference (if wildebeest)
Left boss ______
Right boss ______
Hunted for trophy quality Yes__ No__

Species __________ Date __/__/__

LION

Skull length:__________

Skull width:__________

Species __________ Date __/__/__

LEOPARD

Skull length:__________

Skull width:__________

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B Boss circumference	___	___

Example species:
Impala, Roan, Sable,
Waterbuck, Lechwe, Puku,
Reedbuck, Duiker, Oribi,
Grysbok

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Species __________ Date __/__/__

	Left	Right
A. Length of horn	___	___
B. Boss circumference	___	___

Example species:
Kudu, Eland, Sitatunga,
Bushbuck

Data Completeness

Like the monitoring systems of many projects in developing countries, ADMADE data sets are filled with a good number of ‘holes’ where no data exists. Virtually all of ADMADE’s data is collected in the field, much of it by community members themselves. The challenges of implementing a monitoring system over a vast project area with poorly equipped staff are significant. Given its scant resources it is commendable that ADMADE has been able to develop any kind of monitoring capacity at all.

Although it is heartening to note that in general data collection has and will probably continue to improve with time, there is still a wide range of consistency across different areas and across different datasets. In general, data from the GMAs in the Luangwa basin have been more consistently collected than those in the Kafue region or other parts of the country. This is because the project headquarters, Nyamaluma Institute, is based in the Luangwa Valley, and teams are better able to train and supervise people that area.

Much of the data collected on certain data forms was not entered in to the database prior to 1999. Data which is collected by Nyamaluma staff on field visits (e.g., staff records, camp updates) are generally more consistent and reliable than those collected by village scouts. Of the resource monitoring datasets, the safari hunting dataset is more complete and reliable than field patrol data. One notable exception to the patchy datasets is the WCRF financial data. This dataset is quite complete because it is part and parcel of the sale of licenses at NPWS/ZWA headquarters. Quotas and license prices are also fairly complete.

When using data from ADM for analysis of impact or trends, one must be aware of the limitations of the datasets and make allowances for any missing data in the interpretation. Often, you may have to eliminate data from certain units because it is suspect, aggregate based on what is available, or normalize records based on a some measure of availability. To see a list of the data available for each area, check the queries under *Data Availability* on the Main Menu.

Spatial Referencing

Virtually all of the data sets in ADM are spatially referenced. For example, demographic data is linked to specific VAGs, and field patrol and trophy data are tied to 5 km^2 grids, and WCRF revenue is linked to hunting blocks. When analyzing these datasets, the spatial element can have a significant influence on other variables.

Fortunately, since the inception of its monitoring program, ADMADE has taken steps to ensure that the spatial element of monitoring data is preserved in the datasets. Nyamaluma’s first information management system was designed to work closely with ArcInfo and ArcView GIS software to display data through maps. These maps have been used in the field as feedback for management operations, and to a lesser extent to look for spatially based patterns in larger regions.

The capacity to present the spatial element of monitoring data through maps has been expanded in ADM. In addition to the Reference Map, interactive maps, with user friendly controls to query the database and customize the display of information, are available from the Main Menu. Creating new interactive maps has also been greatly simplified to the point where no specialized GIS or programming experience is needed to make new map definitions. In addition, ADM queries are available to desktop mapping programs such are ArcView which support ODBC. See Linking Access and ArcView for further details on using ODBC.

GMAs, Chiefdoms, Units, Hunting Blocks, and VAGs

For people intimately familiar with the operation of the ADMADE program, the terms Game Management Area, Unit, and Hunting Block can be, and often are, used interchangeably. Indeed for many areas of Zambia these three terms refer to the very same geographic region. However to outsiders, and more importantly to a relational database, these terms need more precise definition.

GMA A Game Management Area is a geographic area, demarcated by the Government of Zambia, which has a special legal status specifying the permitted types of land uses. In ADMADE parlance, GMAs can house one or more ADMADE Units and one or more hunting blocks. These hunting blocks and units will often have the same name as the GMA, but are much different in terms of administrative function and data collection.

In the ADMADE Data Manager, no dataset is entered according to GMA, and hence they are not particularly significant except as a general reference term.

Chiefdom Chiefdoms are the boundaries of the traditional authorities. In the rural areas of Zambia where ADMADE operates, Chiefs maintain a considerable amount of authority. During ADMADE’s first 10 years, Chiefs had a significant impact on the success of the program, both good and bad. Although the role of Chiefs was changed to the mostly honorary position of patron in 1999, they remain quite instrumental in ADMADE activities.

Often, the name of a Unit or Hunting block is the same as the name of the local Chief, and they may be used inter-changably. For example, Lower Lumimba is often called Mwanya after the presiding chief. However no datasets in the ADMADE Data Manager are directly tied to a chiefdom. Nevertheless the interpretation of data from the ADMADE Data Manager is often enhanced with knowledge of the history of the area and the Chief.

Hunting Block A Hunting Block is an area which is designated for legal hunting. Usually a single hunting block will fall within a single GMAs under the management of a single Unit, but some large GMAs have multiple hunting blocks. There are also a few examples in depleted areas of a single hunting block being composed of non-contiguous areas across multiple GMAs.

A single hunting block which has enough wildlife to support hunting is “rented” by NPWS to a single operator, who pays a hefty concession fee for the privilege of hunting in that area. The operator will then contract a professional hunter to maintain a safari camp in the hunting block and guide safari clients.

In the database, safari hunting records (e.g., trophy size) are pegged to a hunting blocks, and financial data from the WCRF database (i.e., license sales) is entered according to hunting block. Although not all hunting blocks actively support hunting, they may still be listed because either (1) there are past safari hunting monitoring records for the hunting block, or (2) the WCRF database has records for the hunting block.

ADMADE Unit An ADMADE Unit, or simply Unit, is an administrative body which manages the wildlife within one or Game Management Areas. A Unit is usually defined by the presence of a Unit Leader, one or more scout camps, support staff, a separate base map, and data forms. Most Units are responsible for only one GMA, or frequently a fraction of a GMA, but there are a couple of examples where multiple GMAs and/or multiple hunting blocks are managed by a single Unit.

With the exception of safari hunting data which is entered according to hunting block, the majority of records in the ADMADE Data Manager are entered by Unit. These include the data on field patrols, staff, and camps.

VAG A Village Area Group is an area within an ADMADE Unit. VAGs are the most recent addition to the ADMADE geographic hierarchy, and were established to help ensure more equitable distribution of benefits and representation on management committees. A Unit will typically have 3-5 VAGs, but some have more.

As implementation of ADMADE in the field continues to evolve, it is envisioned that VAGs will play a greater role in resource management and providing community services. Subsequently move records in the database will be entered according to VAG. These include demography and community projects.

Shifting Boundaries and Relationships

Unfortunately, at least for the designers of databases, there is no simple relationship between hunting blocks, GMAs, and Units which fits all cases. In some cases, there is a one-to-one relationship between Units and Hunting Blocks, and in others there is a one-to-many (meaning one Unit administered more than one hunting block). Likewise some Units, and even some hunting blocks, don’t fit neatly into one GMA. To make matters even more confusing, these relationships can change over time when Units or hunting blocks split into two. Even the relationship between VAGs and Units will likely change when one Unit splits into two or VAGs boundaries are redrawn.

In ADMADE’s old information management system, these shifting relationships were not much of a problem because data was stored in different directories for each area and then each year, and the operators knew the relationships from personal experience. However in a relational database system like Access, whereby information for all areas and all years is stored under one data structure, this type of inconsistent pattern of relationships in the top lookup tables is a nightmare, and can result in significant errors. Trying to capture these relationships accurately would require the unwieldy mechanism of creating three many-to-many relationship tables and incorporating a temporal field in the primary key index.

Fortunately, the lack of a universal and consistent relationship between GMAs, hunting blocks, Units, and VAGs does not significantly impinge on the operation of the ADMADE Data Manager. Most outputs (i.e., reports, queries, maps) from the database can be created and filtered based on one of these three primary lookup tables. For example, a query summarizing safari hunting records can be based on hunting block and year, and a report of field staff can be filtered on Unit. The only potential problems will come when you want to do an analysis which mixes and matches datasets which are linked to different lookup tables. For example, if you wanted to make a query comparing revenue data from the WCRF database, which is entered according to hunting block, with community development projects, which are entered according to Unit, over the past 10 years, you would have to utilize a many-to-many-with-time relationship table to match the two together. This is not an impossible data structure to contend with, but requires careful planning.

Chapter 2: Using ADM

The ADMADE Data Manager has been designed to be easy to use for people who are not familiar with Access 97 and to remove much of the drudgery of performing everyday tasks, such as entering data and printing reports. However at the same time it still preserves all of the functionality of a powerful relational database system such as MS Access, and can be easily expanded to incorporate new objects and data.

This chapter describes some of the features that make ADM easy to use and the tools it provides for managing and analyzing data from a project like ADMADE.

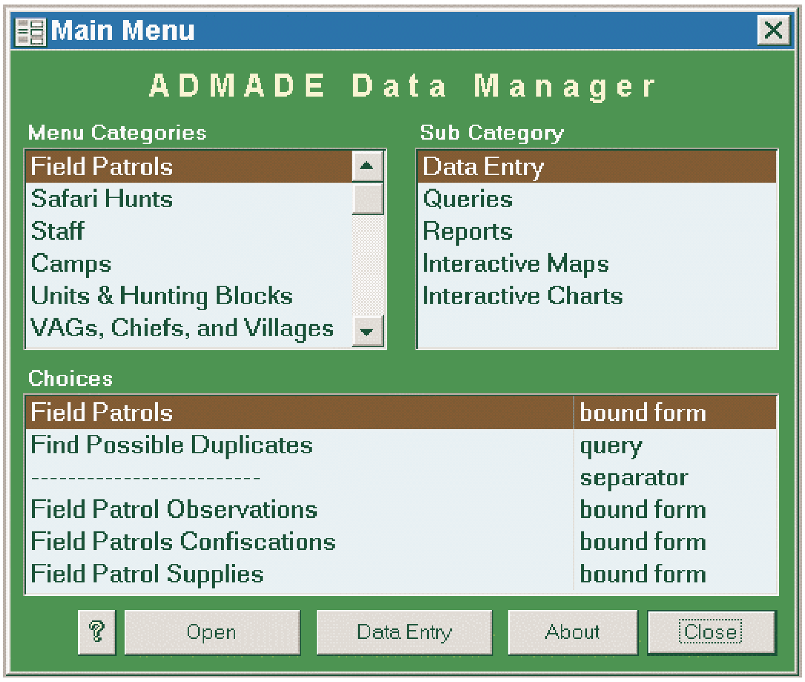
The Menu System

The ADMADE Data Manager has an easy-to-use and easy-to-expand menu system. This menu system integrates program navigation (i.e., opening forms, queries, and reports), along with filtering and documentation.

Main Menu

The Main Menu appears when ADMADE Data Manager first opens, and is a classic ‘switchboard’ type menu that give you access to all of the objects in the database.

To use the Main Menu, select a menu category, then sub-category, and then finally the form, report, or query you want to open. Then click one of the buttons at the bottom.



Main Menu

ADMADE Data Manager

Menu Categories

- Field Patrols
- Safari Hunts
- Staff
- Camps
- Units & Hunting Blocks
- VAGs, Chiefs, and Villages

Sub Category

- Data Entry
- Queries
- Reports
- Interactive Maps
- Interactive Charts

Choices

Field Patrols	bound form
Find Possible Duplicates	query
------------------------	separator
Field Patrol Observations	bound form
Field Patrols Confiscations	bound form
Field Patrol Supplies	bound form

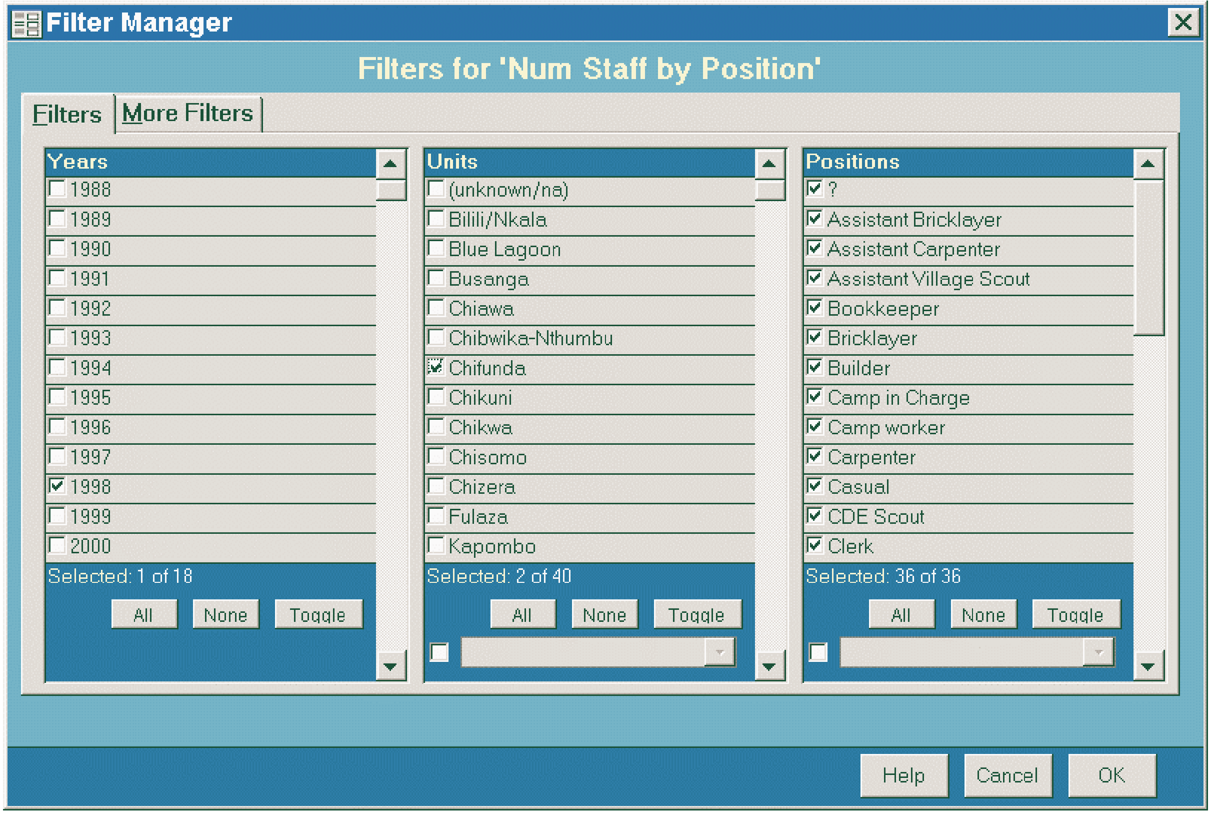
The Main Menu also has a Reminder Message feature, which you can use to schedule messages to appear on certain dates. If you see the Bell icon on the main menu, it means there is a reminder message for that day, which you can see by clicking on the bell.

Note that when the ADMADE Data Manager is first opened, the database window is hidden and only the main menu is visible. Closing the Main Menu will unhide the database window, or you can press F11.

See also: Adding a New Object to the Menu System

Filter Manager

Many of the forms, reports, and queries in ADM have been set up to work with the Filter Manager to give you an easy way to select which records should be included in the output of the object. When you open one of these objects from the Main Menu, you first see the Filter Manager whereby you can make your selections.



Filter Manager

Filters for 'Num Staff by Position'

Years	Units	Positions
□ 1988	□ (unknown/na)	☑ ?
□ 1989	□ Bilili/Nkala	☑ Assistant Bricklayer
□ 1990	□ Blue Lagoon	☑ Assistant Carpenter
□ 1991	□ Busanga	☑ Assistant Village Scout
□ 1992	□ Chiawa	☑ Bookkeeper
□ 1993	□ Chibwika-Nthumbu	☑ Bricklayer
□ 1994	☑ Chifunda	☑ Builder
□ 1995	□ Chikuni	☑ Camp in Charge
□ 1996	□ Chikwa	☑ Camp worker
□ 1997	□ Chisomo	☑ Carpenter
☑ 1998	□ Chizera	☑ Casual
□ 1999	□ Fulaza	☑ CDE Scout
□ 2000	□ Kapombo	☑ Clerk
Selected: 1 of 18	Selected: 2 of 40	Selected: 36 of 36

To use the Filter Manager, you simply put a check in the boxes of those items you wish to be included in the output. For example, suppose you open a query which lists the number of staff at each Unit. The Filter Manager for this query would show probably show a list of all the Units, as well as Years and perhaps staff positions. By checking the appropriate boxes, you can make the query calculate the number of staff for all Units, or just a couple. By putting check marks in the years list, you can select which years should be included. And by making selections in the Staff Positions list, you can tell it to count only the number of staff in the selected positions.

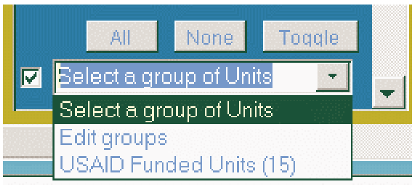
For reports on the menu system, the Filter Manager also provides a space for a custom page header. The user can accept the default or change the report page header for the specific purpose.

The filters which you see in the Filter Manager depend on the form, report, or query being opened. If you were printing a report of the demography of VAGs, you wouldn’t need to see the list of wildlife species. The person who created the form, report or queries and added it to the Main Menu is the one who decides which filters should appear on the Filter Manager for that object. Note that although you can only see at most three lists at a time, the Filter Manager can present up to 8 lists for filtering at one time. To see any additional lists, click the More Filters tab.

Some of the lists on the Filter Manager have *All*, *None*, and *Toggle* buttons which allow you to automatically select all items in the list, unselect all items, or toggle the selection of items.

The All, None, and Toggle buttons let you quickly select items from a list

In addition, some of the lists on the Filter Manager let you define groups of items to make them easy to reselect. For example, the list of Units has a combo box which you can use to choose a group of Units. (Note before the combo box of groups can be opened, you must first check the box next to it). Once you select a group of Units, then the All, None, and Toggle buttons only apply to the items in the select group. To add, delete, or edit groups of items, double-click on the combo box.



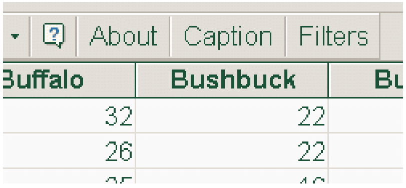
Once a group of Units is selected, the All, None, and Toggle buttons only change the items in that group

Suppose, for example, that you have defined a group of 15 Units which are funded by USAID, and you want only those Units to be selected in the Filter Manager. To quickly select these Units after the Filter Manager opens, you would:

1. Click the None button (to unselect the current selection)
2. Click the check box next to the group combo box (to enable the combo box)
3. Select your pre-defined group of Units from the drop down combo box
4. Click the All button (to select all Units in that group)

Thus with only 4 mouse clicks you have selected your group of Units.

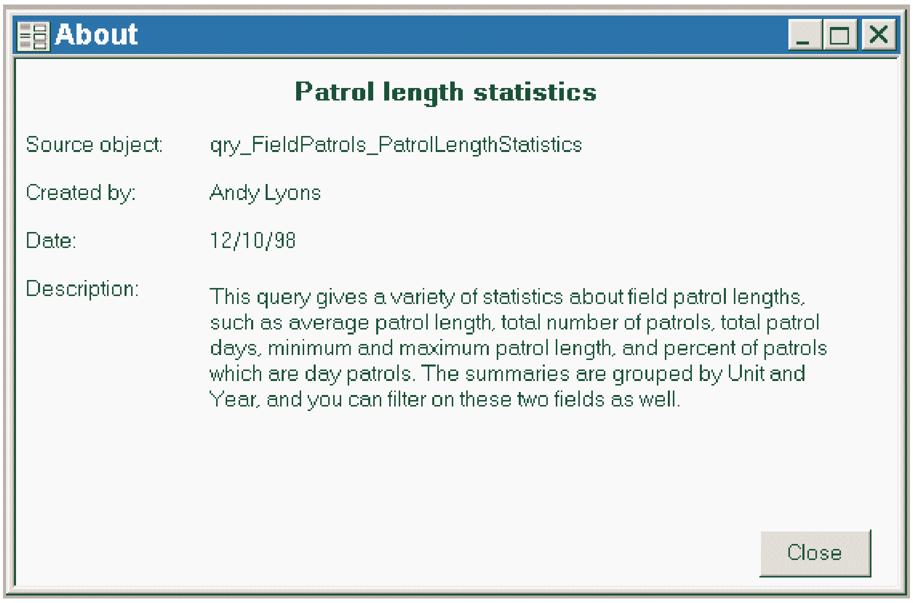
The Filter Manager can also be opened *after* a form, report, or query has been opened. To do this, click the Filter button on the form or toolbar. you can then make new selections, and refresh the object.



The About window, Caption, and Filter Manager can also be opened ***after*** *a form, report, or query is opened by clicking a button on the toolbar.*

About Window

The About Window, which can be opened from either the Main Menu or a button on the query, form, or report, shows basic information about an object. This is part of the built-in documentation system of ADM, and tells you who created the object, when, and what it does. You can also open the About window by right-clicking on a menu choice in the Main Menu.



Patrol length statistics

Source object: qry_FieldPatrols_PatrolLengthStatistics

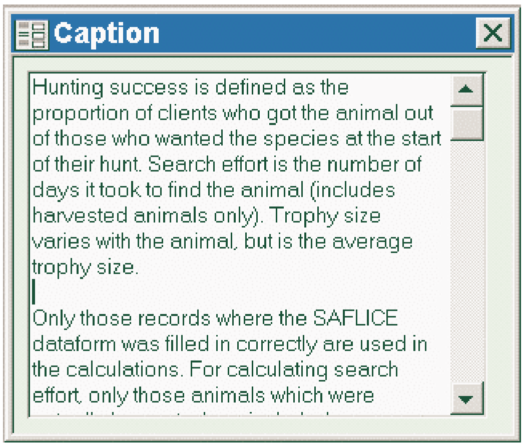
Created by: Andy Lyons

Date: 12/10/98

Description: This query gives a variety of statistics about field patrol lengths, such as average patrol length, total number of patrols, total patrol days, minimum and maximum patrol length, and percent of patrols which are day patrols. The summaries are grouped by Unit and Year, and you can filter on these two fields as well.

You enter text for the About Window when adding an object to the Menu System using the Menu Manager.

Caption Window



Hunting success is defined as the proportion of clients who got the animal out of those who wanted the species at the start of their hunt. Search effort is the number of days it took to find the animal (includes harvested animals only). Trophy size varies with the animal, but is the average trophy size.

Only those records where the SAFLICE dataform was filled in correctly are used in the calculations. For calculating search effort, only those animals which were

Some objects, particularly the interactive graphs and maps, require more description than can be provided by the About window. The Caption window is designed to give the author a place to provide a lengthier description of what the graph, map, or query represents.

The Caption window can also list the current status of the lookup tables the object is filtered on. For example, if you're looking at a map which shows the number of animals in killed in each grid, the Caption window may list the species and years which are being summed to produce the map.

The text in the Caption window can be copied to the clipboard by highlighting it and pressing Control-C. It can then be pasted into another Windows application such as MS Word or PowerPoint. This can be handy when you're using ADM to make graphs or maps for a report or presentation.

See also: Adding New Objects to the Menu System

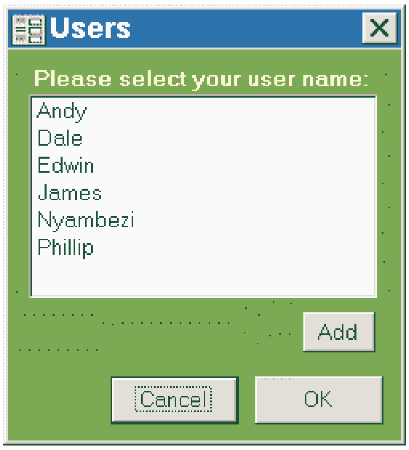
Data Logging

ADM supports behind-the-scenes data logging, which is simply the keeping track of changes made to the data. When data logging is enabled, ADM will make an entry in the data log each and every time a user adds, edits, or deletes a record from a table.

The main purposes of data logging are to:

- help re-enter data in the event of data loss or file corruption (see also Backing Up ADM)
- keep track of users who are editing data and when
- keep track of maintenance on the ADM files, including backing up, repairing, and compacting files
- help resolve synchronization conflicts

When the user first opens a data entry form, they see the window to the right whereby they can select their user name from list. This only has to be done once a session (i.e., until you close ADM and open it again), from then on any changes made to the tables will be attributed to that user.



Please select your user name:

Andy
Dale
Edwin
James
Nyambezi
Phillip

Tip: Double-clicking on a name is the same as selecting it pressing OK.

The following are important points to remember about data logging:

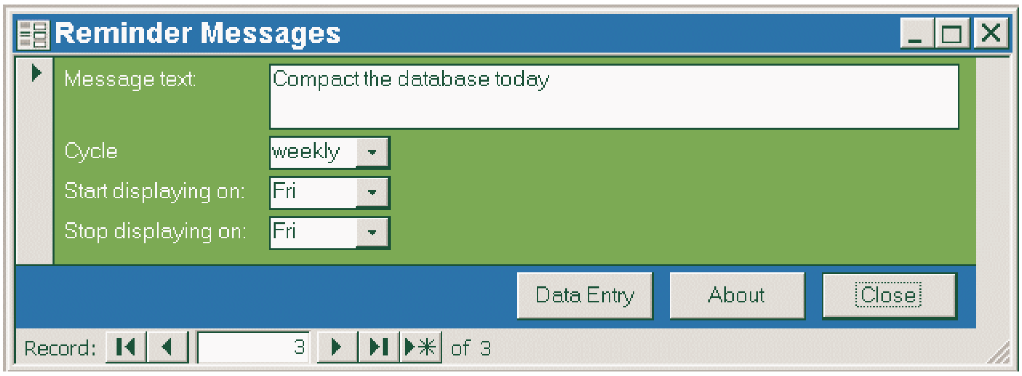
- Data logging can be turned on or off globally (i.e. for the entire program) by disabling it in Program Options (under the Admin menu). It can also be temporarily disabled for a single data-entry form when by clicking Cancel on the Login window. Hence data logging is not something which is enforced. It is up to the user to voluntarily enable data logging and select their proper user name.

- Records are made in the log table only when the user is using one of the data entry forms which has been set up for data logging. If you add/delete/edit records in a table window or query, no record will be made in the Data Log. That's why whenever possible it's better to use the data entry forms to change data. It's easy to incorporate logging on a data entry form, see Adding Data Logging to a Data Entry Form.

- To view entries made in the Log, open the Data Log report from the Main Menu. Alternately, you can open the Edit Data log form to view or print the Data Log.

Reminder Messages

The Reminder Messages feature allows you program messages to pop up on a date you specify. These messages could remind you to performance program maintenance, such as compacting or backing up the database, or do some administrative function such as start preparing graphs for a quarterly report.



Reminder Messages

Message text: Compact the database today

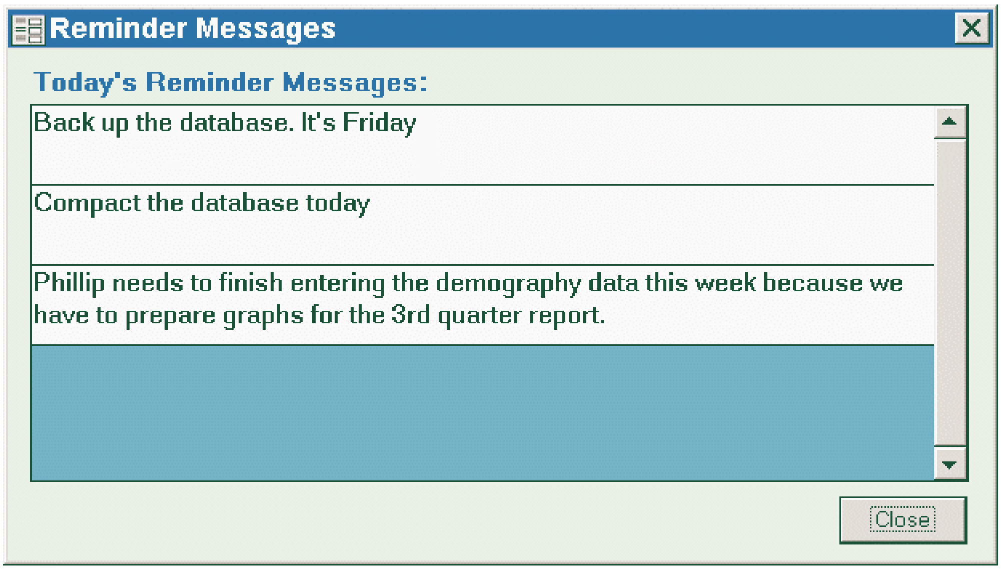
Cycle: weekly

Start displaying on: Fri

Stop displaying on: Fri

There are three display cycles you can choose: weekly, monthly, and yearly. The options for starting and stopping the message display change depending on the display cycle.

Once you enter a reminder message, your message will popup on the specified day(s) when the Main Menu is opened (if this option has been enabled under the Program Options). Also your message can be seen by clicking the bell button on the Main Menu, which only appears when there are messages for that day.


Reminder Messages

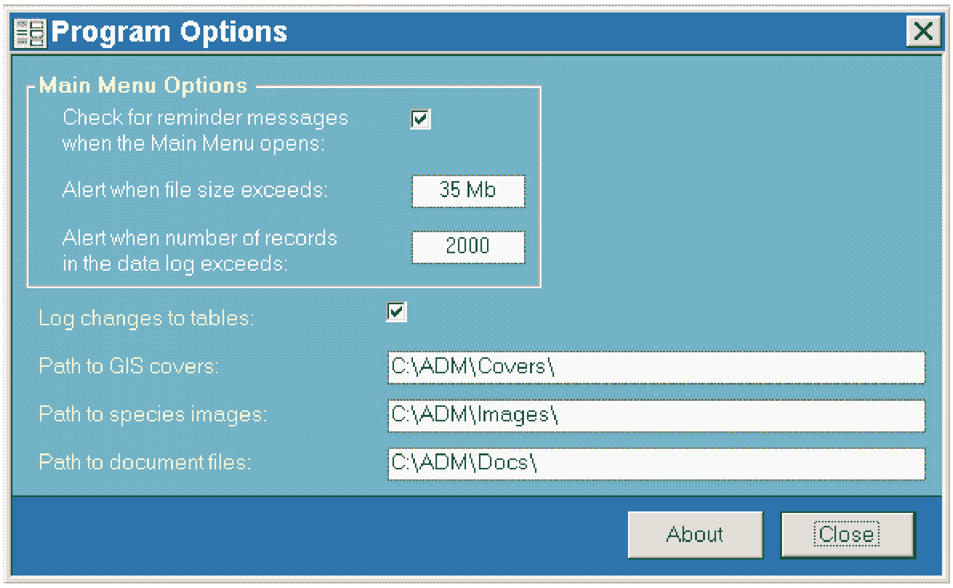
Today's Reminder Messages:

Back up the database. It's Friday

Compact the database today

Phillip needs to finish entering the demography data this week because we have to prepare graphs for the 3rd quarter report.

Program Options


Program Options

Main Menu Options

Check for reminder messages when the Main Menu opens: ☑

Alert when file size exceeds: 35 Mb

Alert when number of records in the data log exceeds: 2000

Log changes to tables: ☑

Path to GIS covers: C:\ADM\Covers\

Path to species images: C:\ADM\Images\

Path to document files: C:\ADM\Docs\

The Program Options dialog (under Admin category on the Main Menu) lets you select whether you'd like certain application features enabled. As can be seen in the form, you can select whether you'd like the Main Menu to check for Reminder Messages each time it opens, and whether you want data logging to be turned on or off for all data entry forms.

You can also set when the Main Menu should show alerts about the size of the file or the number of records in the data log. These alerts can help you determine when a database needs to be compacted, which is particularly important in a replicated database. If there are too many records in the Data Log, this table can be archived to an external file and/or cleaned of old records to reduce file size. If the replica file itself is too big, it can be compacted. By setting either of the alert limits to "0", that alert feature will become disabled.

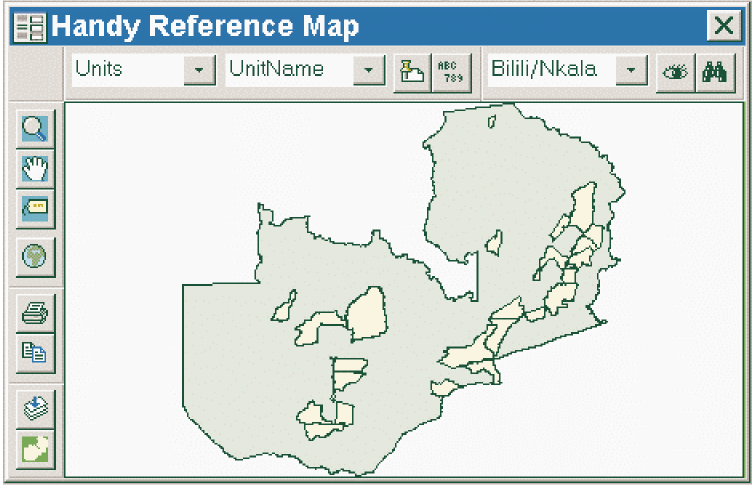
The Program Options dialog is also where you enter the correct path to the GIS covers, images, and document files. This may have to be done if you've set up ADM on a new machine and are using a directory tree structure which is different than the default.

Note that settings from the Program Options are saved in the Windows registry, as opposed to an Access table. Hence preferences set by users won't be altered when the files are synchronized.

Handy Reference Map

Because so much of ADMADE's data in is spatially referenced, the ADMADE Data Manager comes equipped with a Handy Reference Map which can be opened at any time to look for scout camps, units, hunting blocks, etc. The Handy Reference Map is available from the Window menu (NOT the Main Menu), and can be kept opened even when other forms or queries are open. It will stay open until closed by the user.

Click on the buttons for a description of their function



Note: the Handy Reference Map requires that Map Objects and the appropriate GIS shapefiles be installed on the computer. See Installing ADM on a New Computer.

Select Layer
This combo box shows the layers which are currently visible. You have to select a layer before you can add labels or find features.
Add/Remove Labels
To add or remove labels for the current layer, click the Label Layer button. To set the label font or size, click the Label Options button.
Label Options
This combo box shows the field(s) for the current layer which can be labeled and/or searched
Blink and Find Buttons
The eyeball button blinks the feature currently selected in the combo box, while the Find button will take you there.
Zoom Tool
The magnifying glass lets you zoom to an area. To zoom in, drag a box or just click on the map. To zoom out, shift-click or right-click on the map.
Pan Tool
The pan tools lets you push the map around in the window.
Identify Tool
The identify tool lets you identify features for the layer selected in the combo box above.
Full Extent
To zoom out to the maps full extent, click the globe button.
Print/Copy Buttons

The print and copy buttons have their normal functions. You can paste your map into programs like MS Word.

Other Layers

The Layers button lets you set the visibility of other available layers, such as roads, rivers, scout camps, etc.

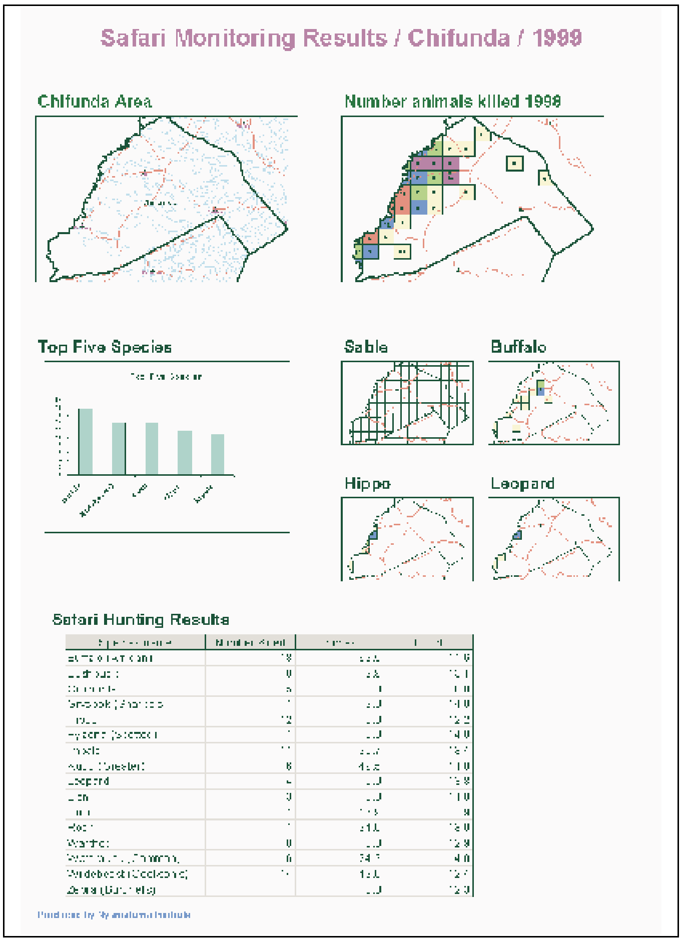
Small Reference Map

This button will open a small reference map which indicates which part of Zambia you're currently looking at.

PowerPoint Slide Templates

Using space-age OLE Automation technology, ADM has the ability to automatically create slides in PowerPoint based on a “template” which you define in Access. Slides created by Access can contain a mixture of maps, charts, tables, and text. When you create the slide, you can make selections to determine which data is reflected. For example, using a Slide Template for hunting results, you could create separate slides for different hunting blocks and years.

Once you have created a PowerPoint slide using ADM, you can use the powerful editing features in PowerPoint to make further changes to your slide. For example you can move elements around, add pictures or additional text, alter colors, or change the dimensions of the slide. You can then incorporate the slide into a presentation, or print it out on a plotter or desktop printer.

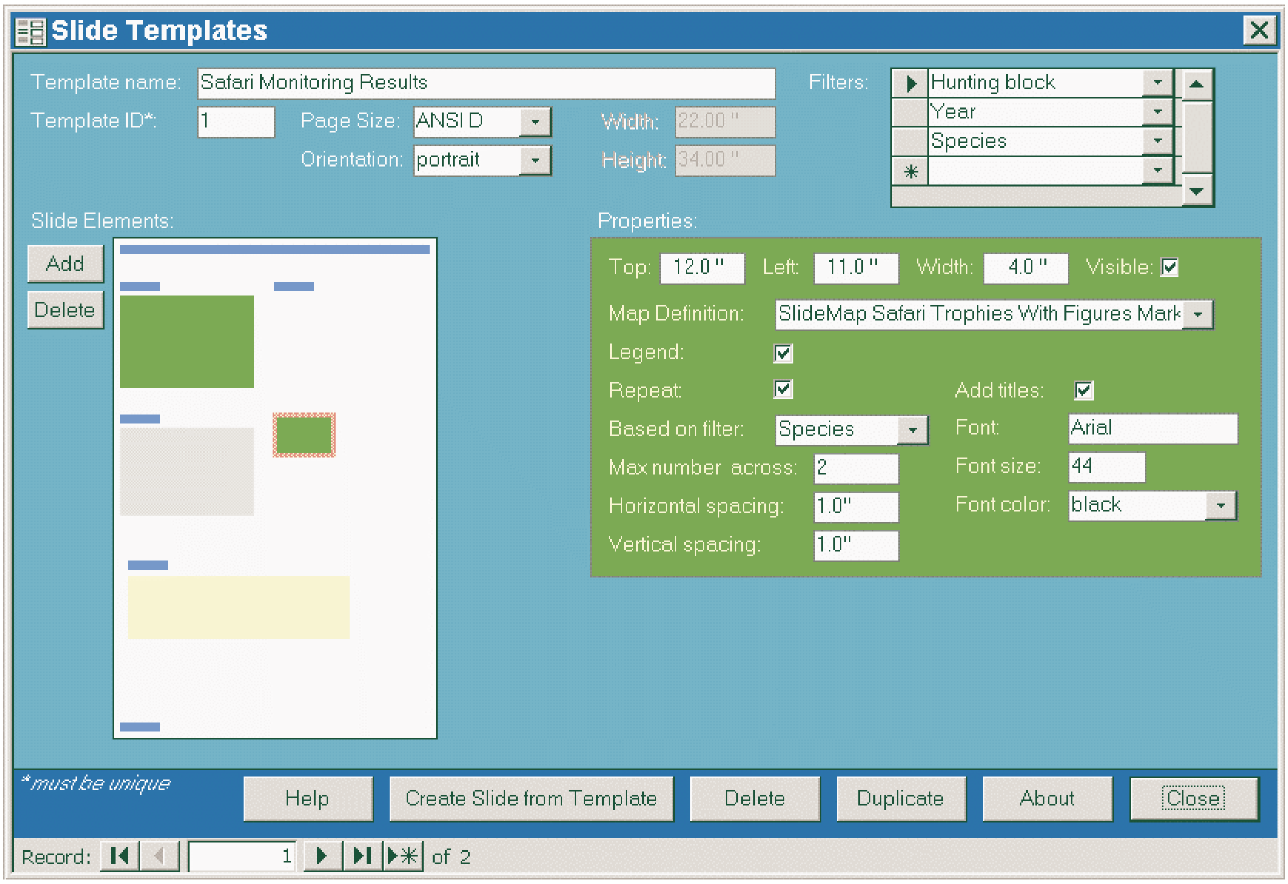


With Slide Templates, you can combine text, charts, maps, and tables in one layout

Combining the analytical processing capabilities of Access and the graphic tools of PowerPoint offers a powerful tool for customizing presentations of data. And because slides can be created with the click of a button, creating them does not take hours of manual design. The potential of this capability is only limited by one’s imagination.

Using the Slide Templates Editor

You create/edit a Slide Template as well make new PowerPoint slides based on the template with the Slide Template form. Select from the Main Menu *PowerPoint Slides*, *Slide Templates*, then *Create Slide Templates*.



Slide Templates

Template name: Safari Monitoring Results

Template ID*: 1

Page Size: ANSI D

Orientation: portrait

Width: 22.00 "

Height: 34.00 "

Filters: Hunting block; Year; Species

Slide Elements:

Properties:

Top: 12.0 " Left: 11.0 " Width: 4.0 " Visible: ☑

Map Definition: SlideMap Safari Trophies With Figures Mark

Legend: ☑

Repeat: ☑

Add titles: ☑

Based on filter: Species

Font: Arial

Max number across: 2

Font size: 44

Horizontal spacing: 1.0"

Font color: black

Vertical spacing: 1.0"

*must be unique

Record: 1 of 2

The first properties that you should set for the Slide Template are the slide paper size, orientation (portrait or landscape), length and width. These values should be entered in the appropriate spaces at the top of the form. After that, you can begin adding elements to your Slide Template. There are four different types of elements that can be added to a slide: text, maps, charts, and tables. Each element on the Slide Template has location properties, such as width, left, and top position, in addition to specific properties such as color, source object, etc. Once you have added elements and set their properties, you click the “Create Slide From Template” button to actually make the slide in PowerPoint.

Maps, charts, and tables on the slide all reflect numeric data from various tables in ADM. The way you get all the maps, charts, and tables to show the same data (e.g., from the same Unit and year) is to filter them using the Marked field from the lookup tables. Then, if you enter the Filter fields in the box at the upper right, the user will see the Filter Manager before the slide is created and have a chance to select which records from the lookup tables should be included in the output.

For example, suppose you want a Slide Template which will include a map, a chart, and table, showing data about field patrol effort for a specific Unit and specific year. When you design a query which the map will use to show the number of hours each grid has been patrolled, you should create filters using the Marked field in the lookup tables for Unit and Years. Another query, which the chart will use to show the number of patrolling man-days by camp, also has filters based on the Marked field from the Units and Years lookup tables. Likewise a third query, which will be the basis of the table on patrolling statistics in the slide, also uses filters from the lookup tables. If all of your Slide Template objects are filtered based on the Marked fields from the lookup tables, when the slide is created all objects will be showing the same data. The user will have the opportunity to select which Unit(s) and Year(s) should be included by making selections in the Filter Manager just before the slide is created.

Once you understand the concepts, creating or editing a slide template using the Slide Templates form is fairly straight-forward. To start a new Slide Template, click the new record navigation button, or duplicate one of the existing Templates by clicking the *Duplicate* button. The slide-layout area on the left shows a representation of how the slide will look. The blue boxes represent text labels, the dark green boxes are maps, the light green boxes are charts, and the yellow boxes are tables. To see the properties of an element, select it by clicking on it in the slide-layout area.

All Slide Template elements have the following properties:

left the distance in inches from the left hand side of the slide to the element
top the distance in inches from the top of the slide to the element
width the width of the element (optional for text elements)
visible whether you want this element to be added to the slide
repeat specifies if the object should be repeated across the page a certain number of times based on user's selection in a lookup table. When creating the slide, you will get a chance to select which lookup records to use for the repeated object. For example, you may want a chart to print once for each camp selected, or a table or map to repeat once for each species selected. When Access creates the chart, table, or map, it will first clear the Marked value for all records in the lookup table, and then reset them one by one and create the objects. The query that your map, chart, or table is based on, should also used the Marked field from the lookup table as a filter, so that it will also

In addition, each type of element has certain custom properties, as follows:

Text elements

The width property on a text element is optional, as PowerPoint will use as much space as needed depending on the length of the text and the size of the font. However if you want your text to be centered across the page, as for a heading, then you should enter a width which matches the width of the entire page and then select *Center* as the justification.

Text elements can also use the phrase "$VAR" in them, which represents a string variable which will be entered by the user when the slide is created. For example, the content of a text element might be "Hunting Results for $VAR". When the user clicks the *Create Slide From Template* button, before the slide is created they will be asked to enter some text to substitute for "$VAR". Hence if the user enters "Chanjuzi 1998 ", the textbox that is created in PowerPoint will read "Hunting Results for Chanjuzi 1998." And if the user has also selected Chanjuzi and 1998 in the Filter Manager, then the charts, tables, and maps on the slide will also reflect only 1998 data from Chanjuzi.

Map Elements

Entering an appropriate value for the width property is mandatory when adding a map element to your template. The height of the map will be determined automatically. Defining a map element also requires that you choose a Map Definition, and check whether you'd like a legend to appear with the map. Like all Interactive Maps, maps created on slides can be bound to queries to classify features, have labels on them, and zoom to the features selected in the lookup table. By double-clicking on the map definition combo box, you can open the Map Definition form where you define how your map should appear on the slide.

Maps pasted into PowerPoint are static images, and have completely lost their connection to data in Access. However they can still be manipulated graphically. If you resize a map in PowerPoint, or any other object for that matter, remember to hold the Shift key down as you drag one of the corner handles to preserve the aspect ratio. By using the Recolor Picture command in PowerPoint, you can alter the colors of the selected map. Finally, if you select the map and then use the Ungroup command from the Draw menu, you can convert the individual features of the map to PowerPoint objects and delete them or move them around as needed.

Charts

Entering an appropriate value for the width property is mandatory when adding a chart element to your slide, however of course you can always resize the chart once the slide is created in PowerPoint. The only other setting to choose is the Chart definition. Double-clicking on the Chart definition combo box opens the Chart Definition form which you can use to design a new chart.

Unlike maps, which are inserted into PowerPoint as static pictures, charts are pasted into your slide as MS Graph objects. Thus while they are no longer linked to the data in Access, you can still double-click on the chart in PowerPoint to change colors, labels, axes, chart type, gridlines, etc.

Tables

You can add tables to your PowerPoint slides, which allow you to add summaries of the data from ADM. The table is created from an Access query which is run at the time the slide is created. Hence your query should contain the appropriate filters based on the lookup tables. It is recommended that you enter an appropriate value into the width property for your table element, but you can always resize it later. Once tables are pasted into PowerPoint, they are no longer linked to the data in Access.

Chart Definitions

A Chart Definition is used to determine how a chart in a PowerPoint slide created from an ADM Slide Template will appear. You define charts in the *Slide Template Charts* form, which you can open either from the Main Menu (under Slide Templates) or by double-clicking the Chart Definition combo box on the Slide Templates form.

Note: Chart Definitions are not the same as and are not used to create Interactive Charts, which are described in Chapter 3.

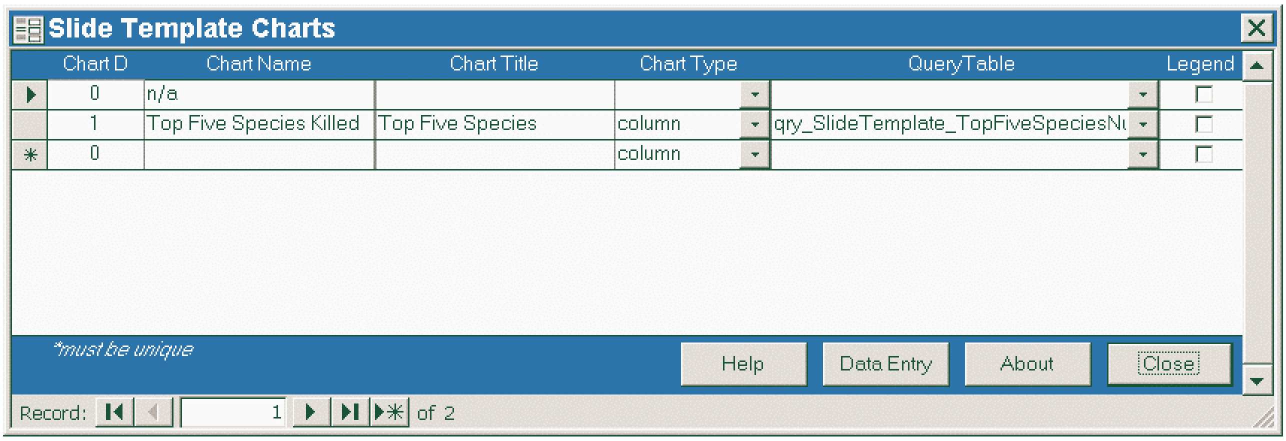


	Chart D	Chart Name	Chart Title	Chart Type	QueryTable	Legend
▶	0	n/a				□
	1	Top Five Species Killed	Top Five Species	column	qry_SlideTemplate_TopFiveSpeciesNı	□
✳	0			column		□

*must be unique

The interface for creating a Chart Definition is fairly straight-forward. The only tricky part about creating a Slide Template Chart is designing the query its based on. Queries used for chart definitions should have their data in rows, not columns. Hence each row should represent data about a particular object, such as a year, grid, Unit, camp, etc. The first column of the query should contain the names of the object. In other words, the year, Unit name, camp name, grid number, etc.

The second column, and any others after that, should contain the numeric values which will be displayed in the chart. Queries for charts should have at least two columns.

Sometimes you only want your chart to display the top 5 or top 10 values. This can be done by setting the Top Values property of your query, which will limit the number of rows returned. Also, remember to filter your query based on the Marked field from the lookup tables, so that the table will correspond with the other elements on the slide.

There are four types of chart types available, column, pie, 3D column, and 3D pie. If your query has more than 2 columns in it and you select colum as the chart type, it will cluster the columns.

You can also enter a chart title and choose whether you want your chart to have a legend. These options and many others can be changed once the chart is created in PowerPoint, by double-clicking on it. In fact, most of the time you will probably want to put some finishing touches on your chart in PowerPoint before it is printed.

Linking Access and ArcView Via ODBC

Although Access 97 is very well equipped to manage text and numeric data, it's abilities to handle spatial data are fairly limited. However fortunately Access and ArcView can work very well together via the Open Database Connectivity drivers.

Most of the data managed by the ADM has a spatial component to it, and some of it, such as grid locations of animal kills, is entirely spatial. You can take advantage of the powerful classification features of ArcView by making Access tabular data available to ArcView via ODBC. The basic steps involved in this are:

1. Create a query in Access which pulls out the tabular data you need in ArcView
2. Open ArcView and import the Access query as a virtual table
3. Join the virtual table to a GIS coverage in the usual manner and create the map display

These steps are described in greater detail below:

Step 1: Create a query in Access which pulls out the tabular data you need in ArcView
The query you create in Access should have the following fields which you will need in ArcView:

Field	**Examples**
a field to join to the GIS cover	grid number, camp_id
one or more fields which contain the numeric data you wish to plot	number of animals killed, number of snares found, total revenue
one or more fields that ArcView will need to filter the records on	year, hunting block, unit

The last field(s) for filtering are particularly important to prevent too much data from being displayed on your ArcView map. The ADMADE Data Manager stores data from all Units for all years, so in your query you need to provide ArcView with field(s) that you can use to select which records to use. For example, suppose you want to create a map showing the total number of animals killed in each grid. You could of course create a separate query in Access for each hunting block and each which tabulates the total animals killed, but that of course would defeat the purpose of using a relational database. A better strategy would be to create query that outputs the following fields:

Hunting Block
Year
Species
Grid
Total Number of Animals Killed

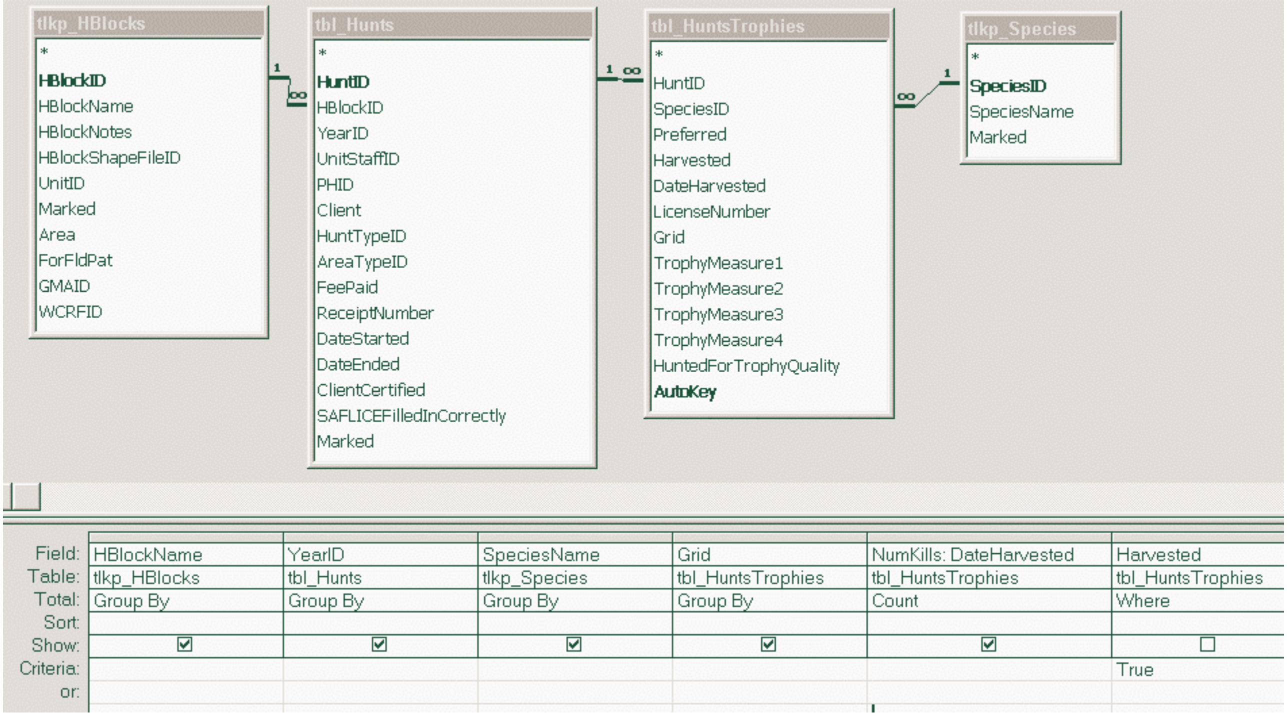
Of course this query is going to have many more records than you're likely to need for one map in ArcView, however you can filter the records when you create your virtual table in ArcView. For example you can create one virtual table for the animals kills in Luawata-1997-Lions, another virtual table for Luawata-1997-Leopard, etc. In this way, one query in Access can be used to supply data for many virtual tables in ArcView, each one based on a different criteria string.

The name of the query you create in Access should begin with 'qav' to designate that this query is going to be used by ArcView. Your query does not need to be put on the menu system, but it should probably be hidden from the database window to prevent accidental modification or deletion.

Another complication you may have to deal with is that Access 97 and ArcView 3.x support slightly different version of Structure Query Language (SQL). SQL is a string which is constructed behind the scenes in Access which defines your query. If you create a query in Access that uses multiple joins, grouping, or functions, ArcView may have a difficult time interpreting the SQL string, and you may get an error when you try to create the virtual table.

To avoid this problem, you can make it a practice of using nested queries in Access. After you create the query which outputs the fields you need in ArcView, save it and then create another very simple 'front-end' query which calls the complex query. For example, suppose you create a query which tabulates the number of animals killed in each grid. In design view, this query would look something like Figure 1.

This query includes all the fields you need in ArcView, but it is too complicated for ArcView to import directly. So you save this query as qav_NumAnimalsKillByGrid_Sub, and then create a new query called qav_NumAnimalsKillByGrid which merely shows all the fields from the first query, as in Figure 2. This query is easy for ArcView to digest and still contains all the fields needed to create the appropriate virtual tables.



tlkp_HBlocks
*
HBlockID
HBlockName
HBlockNotes
HBlockShapeFileID
UnitID
Marked
Area
ForFldPat
GMAID
WCRFID

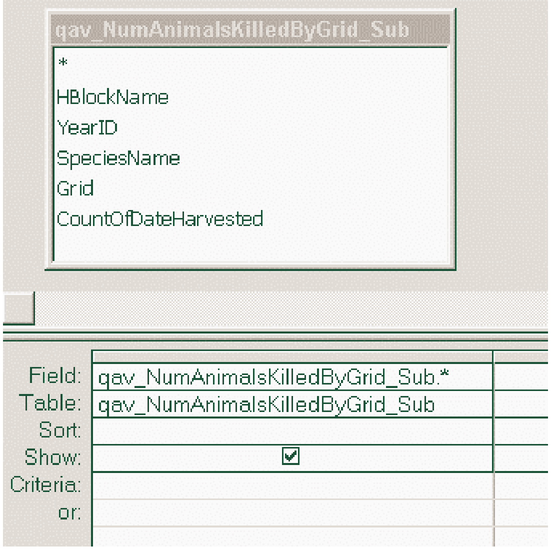
tbl_Hunts
*
HuntID
HBlockID
YearID
UnitStaffID
PHID
Client
HuntTypeID
AreaTypeID
FeePaid
ReceiptNumber
DateStarted
DateEnded
ClientCertified
SAFLICEFilledInCorrectly
Marked

tbl_HuntsTrophies
*
HuntID
SpeciesID
Preferred
Harvested
DateHarvested
LicenseNumber
Grid
TrophyMeasure1
TrophyMeasure2
TrophyMeasure3
TrophyMeasure4
HuntedForTrophyQuality
AutoKey

tlkp_Species
*
SpeciesID
SpeciesName
Marked

Field:	HBlockName	YearID	SpeciesName	Grid	NumKills: DateHarvested	Harvested
Table:	tlkp_HBlocks	tbl_Hunts	tlkp_Species	tbl_HuntsTrophies	tbl_HuntsTrophies	tbl_HuntsTrophies
Total:	Group By	Group By	Group By	Group By	Count	Where
Sort:						
Show:	☑	☑	☑	☑	☑	□
Criteria:						True
or:						

Figure 1 - A somewhat complex query which totals the number of animals killed in each grid by hunting block and year



qav_NumAnimalsKilledByGrid_Sub
*
HBlockName
YearID
SpeciesName
Grid
CountOfDateHarvested

Field:	qav_NumAnimalsKilledByGrid_Sub.*
Table:	qav_NumAnimalsKilledByGrid_Sub
Sort:	
Show:	☑
Criteria:	
or:	

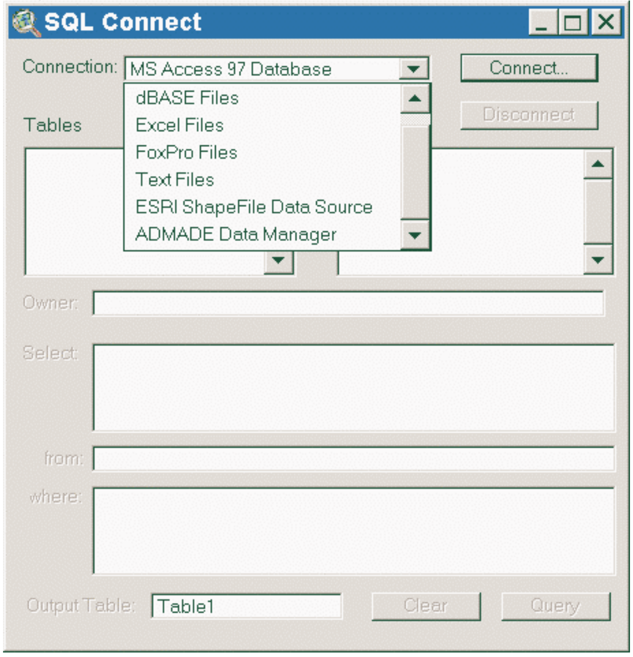
Figure 2 – A very simple query which displays all the fields from the query shown above

Step 2: Open ArcView and import the Access query as a virtual table

ArcView has the ability to create virtual tables which are connected to external databases via ODBC. Virtual tables work and function just like any other ArcView table, however they are linked to an ODBC source and are re-created each time you open the Project.

To bring your Access query into ArcView, select "Connect SQL" on the Project menu from the Project Window. You will then see a list of available ODBC Sources.

Note: Access doesn't have to be running, when you connect to the ADM_MENU.MDB file. In fact since both Access and ArcView are memory hogs, you might have better performance if Access is closed.



Connection: MS Access 97 Database

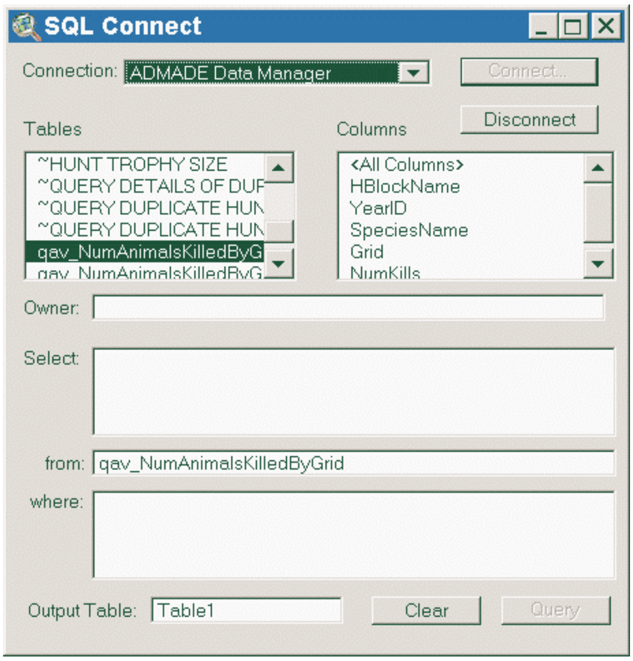
dBASE Files
Excel Files
FoxPro Files
Text Files
ESRI ShapeFile Data Source
ADMADE Data Manager

from:

Output Table: Table1

Note: you won't see ADMADE Data Manager listed as one of the ODBC connection sources unless you have set it up in the Windows 95 control panel (32bit ODBC).

After you select the appropriate connection and click the Connect button, you will see a list of all of the tables and queries in the ADMADE Data Manager. Double-click the qav_NumAnimalsKilledByGrid, and you will then see a list of all the fields in the query. Your SQL Connect window should now look like:



Connection: ADMADE Data Manager

~HUNT TROPHY SIZE
~QUERY DETAILS OF DUF
~QUERY DUPLICATE HUN
~QUERY DUPLICATE HUN
qav_NumAnimalsKilledByG
qav_NumAnimalsKilledByG

<All Columns>
HBlockName
YearID
SpeciesName
Grid
NumKills

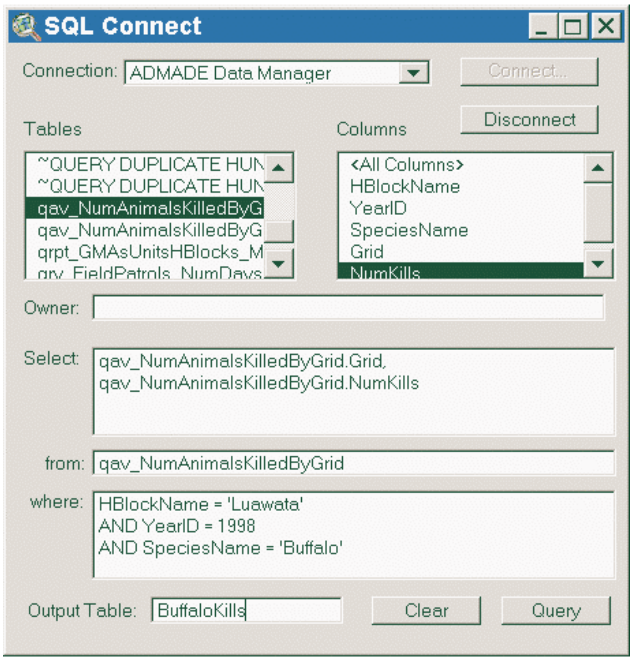
from: qav_NumAnimalsKilledByGrid

Output Table: Table1

Next you need to select which fields you want from the query to appear in your virtual table. You do this by double-clicking on the field names in the Columns box. When you double-click on a field name, it appears in the Select box. You can also type field names directly in the Select box. Note you don't *have* to type in the name of the query when selecting field names.

Next type in a where clause, using standard logical expression syntax. Remember to use AND and OR keywords. Also note that character strings should be delimited with single quotes, not double-quotes.

Finally enter the name for your virtual table, and click the query button. You now have a linked table in your ArcView project.



Connection: ADMADE Data Manager

Owner:

Select: qav_NumAnimalsKilledByGrid.Grid, qav_NumAnimalsKilledByGrid.NumKills

from: qav_NumAnimalsKilledByGrid

where: HBlockName = 'Luawata'
AND YearID = 1998
AND SpeciesName = 'Buffalo'

Output Table: BuffaloKills

Unfortunately, ArcView can not handle one-to-many joins with attribute tables, so in the example above you would have to create one virtual table for each species for each year. However the good news is that you only have to create a virtual table once. Thereafter ArcView will requery the ODBC source each time the project is opened and recreate the virtual table with the most recent data. Also the Import ODCB Table wizard extension in ArcView can be used to create multiple virtual tables from a single Access query based on user selected values in one or more filter fields.

Chapter 3: Expanding ADM

The ADMADE Data Manager is designed to grow and be flexible enough to incorporate other types of data and analyses. Many times, you will simply need to create a new query for a specific purpose, such as a report or presentation. Other times there may be a whole new data set to incorporate, such as transect data or socio-economic data from some survey. Regardless of how you need to expand the ADM, a few points are worth remembering:

1. Try to keep the objects you create flexible so they can be reused over and over for different purposes. For example, suppose you need to create a query to know the population growth for a specific Unit because you're about to take a trip there. You could of course filter the query by entering the name of the Unit as a criteria expression, however it only takes a smidgen more effort to join the lookup tables to your query, filter it using the Marked field from the lookup table, and put it on the Menu System, so that the next time you or someone else needs this information they can use the Menu System and Filter Manager to perform the analysis for a different Unit.

2. You can create new local objects (e.g., queries, forms, reports) in any replica, but you can only add or make changes to replicated objects in the Design Master. The difference between a local object and a replicated object is that replicated objects get copied to the other replicas when you do synchronization.

3. If you're doing a quick and dirty analysis for a very specific project, and don't want to save your queries/reports on the Menu System, then you could start a new Access database and make a link to all the tables in the data file. For example you could start a new Access database called SUSTAIN_RPT.MDB, link all the tables from ADM_REPLICA.MDB, and make all your queries that you need for the report. Later, if you think some of those queries may actually be useful for other purposes, you could import them into the design master and put them on the menu system.

4. Document any queries, forms, and table you create. A big advantage of using the Menu System is that it gives you a place to document what your object does, but even if you can't do that at least make notes of what the query is for. One of the boxes in the property sheet for queries is a Description box, which you can use to write what the object was created for. If you're not sure if you've entered enough description, ask yourself if it will still be clear to you or someone else 2 years from now.

This chapter describes some of the techniques and tools you can use to expand the ADMADE Data Manager.

Keeping the Database Tidy

Access 97 is very good in helping you create queries and reports that you can use to analyze and present data in very powerful and flexible. However with as many interrelated datasets which are housed under the ADMADE Data Manager, these is a very real danger that the database can become cluttered with queries and reports whose purposes are long since forgotten. This can quickly result in a database which is difficult to use, and increase the amount of errors because processes which are opaque to the user, The way to avoid this scenario is to adhere to the following guidelines:

1. Use naming conventions when saving your objects
2. Make use of the ADM Menu System
3. Only replicate those objects (e.g., tables and queries) that need to be copied to all replicas. Other objects should be made local.
4. Periodically clean up objects which are no longer needed.

Naming Conventions

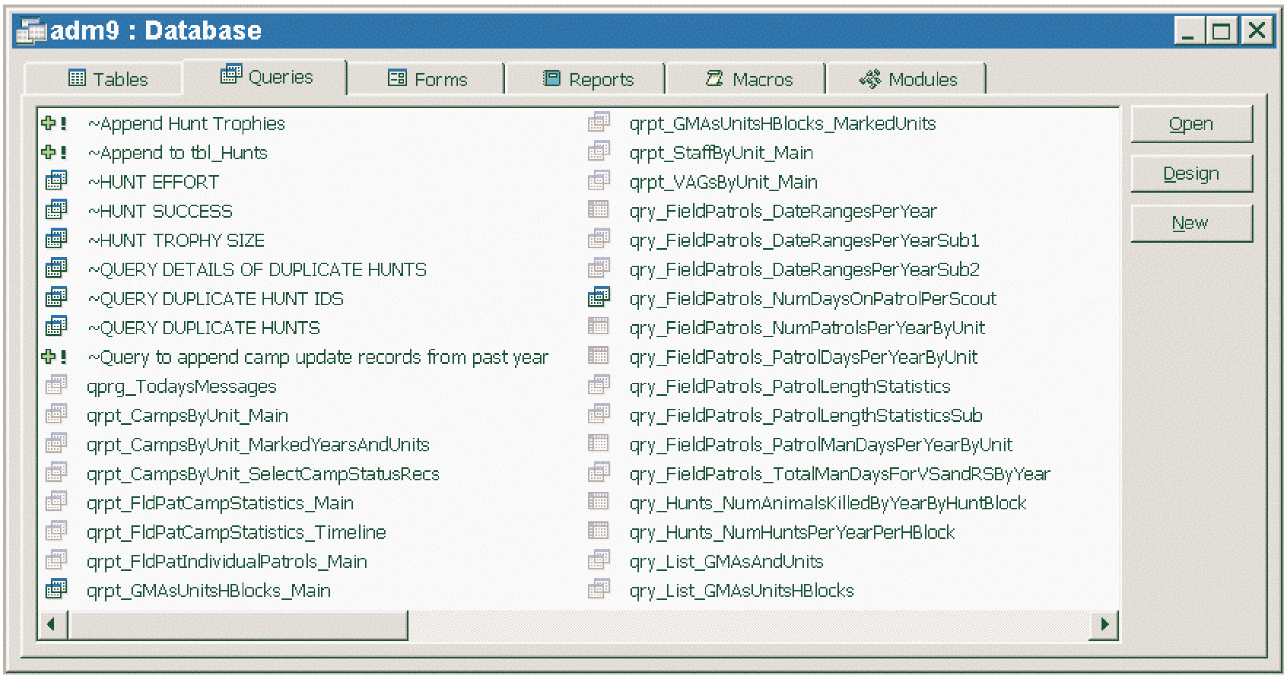
Because Access stores all tables, queries, forms, and reports in one big file, it can be difficult with a large database to find objects in the database window. Fortunately, Access allows you to use up to 64 characters when naming database objects (e.g., queries, reports, tables, forms). By following naming standards you can make certain that related objects will appear together in the database.

Objects in the ADMADE Data Manager should adhere to the industry-wide Leszynski Naming Conventions (LNC), which uses a standard system of prefixes to keep related objects together in the database window. Basically, the name of each object should have three parts:

1. a prefix which tells you what type the object is
2. a stem or root which tells you something of the object's function
3. optionally a suffix which links dependent objects together (such as one query that is needed to run another query).

The best way to introduce these naming standards are with some examples:

Prefix	Dataset + Function	Suffix	Full Name	Description
tbl	FieldPatrols		tbl_FieldPatrols	Main table for field patrol data
tbl	FieldPatrols	Supplies	tbl_FieldPatrolsSupplies	A "details" table which records supplies for each field patrol
frm	CampsUpdate		frm_CampsUpdate	The main data entry form for camp updates
frm	CampsUpdate	Sub	frm_CampsUpdate_Sub	The sub form which contains the detail records of each camp for each year
rpt	FldPatIndividualPatrols		rpt_FldPatIndividualPatrols	A report which shows statistical information about individual field patrols
rpt	FldPatIndividualPatrols	Confiscations	rpt_FldPatIndividualPatrolsConfiscations	One of the sub-reports which lists the details of the confiscations.



~Append Hunt Trophies
~Append to tbl_Hunts
~HUNT EFFORT
~HUNT SUCCESS
~HUNT TROPHY SIZE
~QUERY DETAILS OF DUPLICATE HUNTS
~QUERY DUPLICATE HUNT IDS
~QUERY DUPLICATE HUNTS
~Query to append camp update records from past year
qprg_TodaysMessages
qrpt_CampsByUnit_Main
qrpt_CampsByUnit_MarkedYearsAndUnits
qrpt_CampsByUnit_SelectCampStatusRecs
qrpt_FldPatCampStatistics_Main
qrpt_FldPatCampStatistics_Timeline
qrpt_FldPatIndividualPatrols_Main
qrpt_GMAsUnitsHBlocks_Main
qrpt_GMAsUnitsHBlocks_MarkedUnits
qrpt_StaffByUnit_Main
qrpt_VAGsByUnit_Main
qry_FieldPatrols_DateRangesPerYear
qry_FieldPatrols_DateRangesPerYearSub1
qry_FieldPatrols_DateRangesPerYearSub2
qry_FieldPatrols_NumDaysOnPatrolPerScout
qry_FieldPatrols_NumPatrolsPerYearByUnit
qry_FieldPatrols_PatrolDaysPerYearByUnit
qry_FieldPatrols_PatrolLengthStatistics
qry_FieldPatrols_PatrolLengthStatisticsSub
qry_FieldPatrols_PatrolManDaysPerYearByUnit
qry_FieldPatrols_TotalManDaysForVSandRSByYear
qry_Hunts_NumAnimalsKilledByYearByHuntBlock
qry_Hunts_NumHuntsPerYearPerHBlock
qry_List_GMAsAndUnits
qry_List_GMAsUnitsHBlocks

Temporary Queries
Queries or tables that have just been created for a temporary purpose and won't be needed in the future should start with the tilda ~ character. These objects will always appear first in the database window, and this tells you that it can be safely deleted.

Report Queries
Because these queries have the prefix 'qrpt_', we know that that they are record source for reports. If someone wants thinks there might be an error in the data in a report, they know just where to look

Stem
The middle part of a query name tells you (i) the primary dataset it uses, and (ii) what specifically the query presents. In this section, all the queries dealing with the Field Patrol data are grouped together, and we can tell from their names what they represent

Hidden Objects
Queries with icons that are dimmed are normally hidden in the database Window. Once an object is on the main menu it can be hidden in the database window to prevent accidental tampering or deletion. This is another way of keeping the database window tidy. We only see hidden queries here because View Hidden Objects is enabled (set under Tools... Options)

Dependencies
We can tell that these three queries go together because their names only differ by the final suffix: Sub1 and Sub2. If there is ever a problem with this query, we know immediately where to start debugging.

Table Prefixes

tbl Tells you that this table is a base table. In other words this table contains data that changes or is updated on a regular basis. Most information that is entered by users will go in a base table. Base tables often contain fields which link to lookup tables. For example there may be a numeric field for the Unit which is linked to the Unit lookup table. Example: tbl_FieldPatrols.

tlkp Signifies a lookup table. Lookup tables contains values used by other tables, and generally contain information that never changes (such as the name of a hunting block and its area). For example, instead of saving the word “Chanjuzi” for every record in the field patrol data table, we store a number which corresponds to a record in the tlkp_Units table. Using lookup values vastly improves the performance of Access, reduces the size of the file, and allows powerful filtering based on lookup tables.

tprg This is a system table used by the ADMADE Data Manager Visual Basic routines. For example all of the menu choices from the menu system are saved in tprg_MenuChoices. Users don’t normally need to look at or modify these tables, and in fact most of them are hidden.

Query Prefixes

qry This is a standard prefix for a standalone query. Most queries will have this prefix.

qrpt This prefix indicates that the query is used as the record source for a report.

qfrm This prefix designates a query which is used as the record source for a form. However you won’t need many of these queries because Access 97 forms have the ability of saving their record source directly with the form.

qav This prefix indicates a query which was created to be used by ArcView. See Linking Access and ArcView Via ODBC.

qmap This prefix indicates the query is being used as part of a Map Definition.

Report and Form Prefixes

rpt and **frm** are the standard prefixes for form and report objects respectively. If the object is used as a sub-form or sub-report (for example to show records from a details table, such as field patrol supplies), that should be indicated by using a suffix.

It may be helpful to give each report some sort of ID number or code, such as R1, R2, etc. This could help managers communicate with technical staff on which report they want to see.

Naming Fields

Field names in tables can be up to 64 characters long, but should not contain punctuation characters. Field names should be descriptive and unique. When assigning field names, think about how that table will be used when creating queries. For example, “Name” is a very common name for a field, but is not very helpful when you’re creating a query that links 5 lookup tables, each of which contains a field called “Name”. Better to call it “UnitName” or “VAGName”, so that no matter where this field appears you will know what it stands for.

Likewise, the names of numeric fields which link to lookup tables should end with “ID”, for example “UnitID”, “HblockID”, etc. This tells you immediately that this is the field which should be used to create joins to lookup tables. Note that Users never need to know or even see the actual numeric values of a lookup table if you use combo boxes to display the field.

Developing Naming Styles

Whether to use spaces, underscores, or hyphens when you give names to tables, fields, queries, reports, or forms, is largely a matter of personal preference. However there are certain advantages of NOT using spaces when you assign names. When creating queries or referring to objects in Visual Basic code, object names with embedded spaces must be enclosed with square bracket characters [ ]. This becomes tedious very quickly.

Using Lookup Tables

Lookup tables are used throughout ADM to improve performance, reduce storage space, and add functionality. Lookup tables typically have at least two fields, a numeric ID number which uniquely identifies each record, and a text field which describes the record in plain English (e.g., Unit name, Species name, etc.). Some lookup tables may also contain other fields containing information about a record which will never change, (e.g., area, province, GMA, etc.).

The use of lookup tables is highly recommended in relational databases such as Access because they (1) greatly reduce the amount of hard disk space required, (2) improve performance, and (3) allow you to take advantage of Access's ability to enforce referential integrity behind the scenes (described below). For example, the table which stores quota data has literally thousands of records, each one containing fields for a species, hunting block, year, and the number of animals on the different types of quotas. It would be grossly inefficient to save the name of the species as a text string such as "Buffalo" in each record. However by assigning each species a unique ID number (in the lookup table), the amount of hard disk space used by the quota table is vastly decreased and the speed of queries using this table increased.

Separating your data into different tables also allows you to use Access's tools to enforce the integrity of the data. For example, suppose you have a details table which has a field called UnitID. Values in this field must correspond to the ID number of a Unit. When you define the relationship between these two tables in Access (under Tools => Relationships), you can tell it to enforce referential integrity such that there can be no record in the details table that doesn't contain a value in the UnitID field that doesn't have a matching record in the Units lookup table. Likewise, when referential integrity is enforced, Access will not allow the user to delete records from the lookup table if there are still records in a related details table that are related to that record. See the Access help system for more information on enforcing referential integrity.

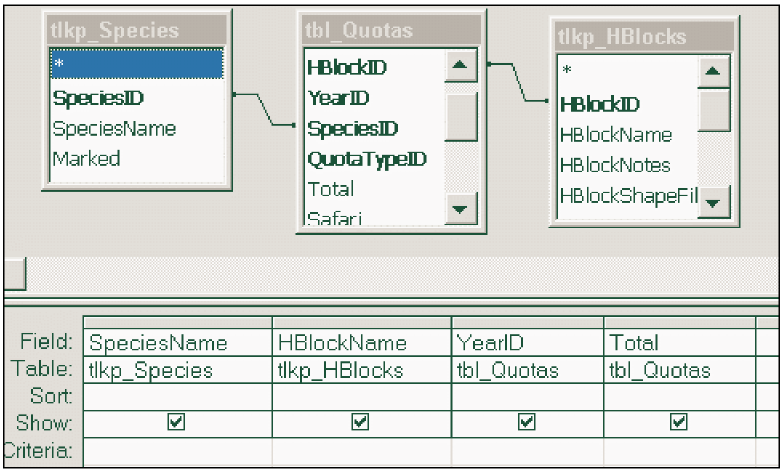
Most lookup tables in ADM have an additional field called 'Marked'. The Marked field is a simple Yes/No field, which means that it's value can either be True (Yes) or False (No). The Marked field is different than the other fields in the lookup table in that it does not represent any meaningful attribute information about the record (such as the Unit name). However the Marked field is quite handy because it can be used to create very powerful filtering capabilities for any query, form, report, interactive graph or map which uses the lookup table. This is described further in: Filtering Queries With the 'Marked' Field from Lookup Tables.

Filtering Queries With the 'Marked' Field from Lookup Tables

Access queries provide extremely useful information, and are the building blocks of almost every other object in Access, including reports, forms, interactive graphs, etc. Mastering queries is the keystone to unlocking the powerful analytical features of a relational database like Access. Queries provide the ability to view data which is:

- from one or more related tables
- from other queries
- sorted according to one or more fields
- summarized using an function such as Average, Sum, Max, etc.
- filtered on one or more fields

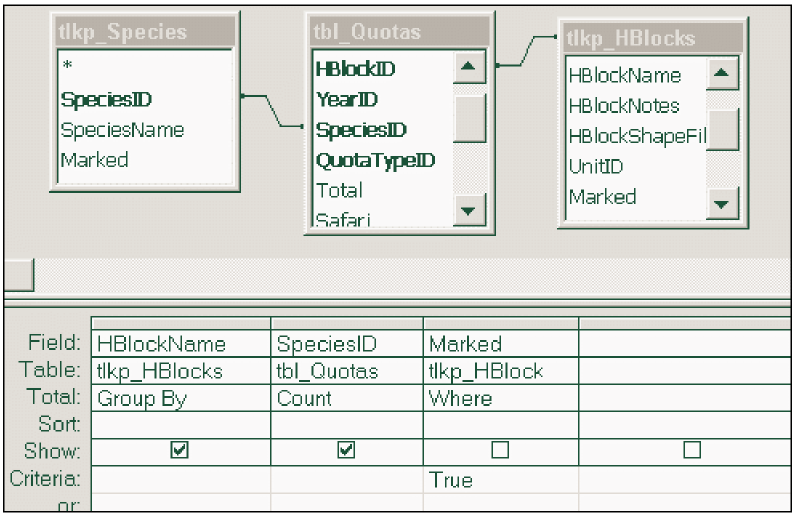
Most of the core data tables in ADM use efficient numeric values linked to lookup tables instead of text fields for information such as the Unit, Species, Supply Item, Confiscated Object, etc. Hence most of your queries will require joining one or more lookup tables to retrieve the more readable text description of the record. For example, if we're creating a report that shows the hunting quotas, then the query for the report needs the species lookup table so we can see the name of the species in the report and not just the ID number.



Field:	SpeciesName	HBlockName	YearID	Total	
Table:	tlkp_Species	tlkp_HBlocks	tbl_Quotas	tbl_Quotas	
Sort:					
Show:	☑	☑	☑	☑	
Criteria:					

This query uses the species lookup table so that the output datasheet will contain the name of each species and not just the ID number

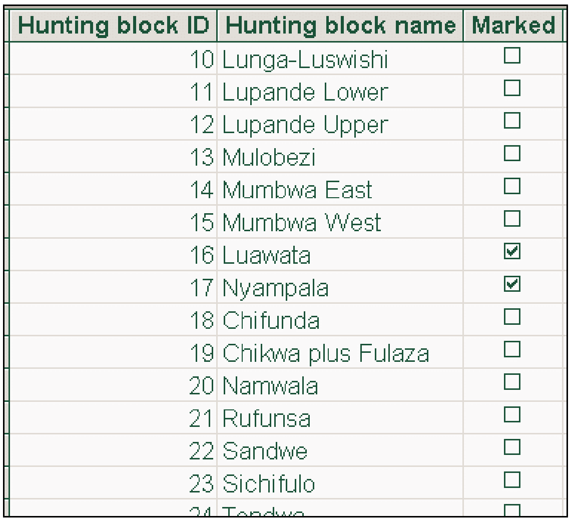
A second advantage of adding lookup tables to your queries is that this allows you to use the Marked field from the lookup tables to restrict (i.e., filter) the output. When you create a query which includes the Marked field from a lookup table in the query grid, and you type "True" as the criteria expression for the Marked field, then only those records from the data table where the matching record in the lookup table has the Marked field equal to True will be in the output. This can be seen below:



Field:	HBlockName	SpeciesID	Marked	
Table:	tlkp_HBlocks	tbl_Quotas	tlkp_HBlock	
Total:	Group By	Count	Where	
Sort:				
Show:	☑	☑	□	□
Criteria:			True	
or:				

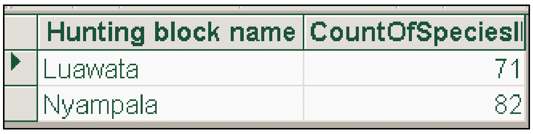
The output of this query will only display records from the quota table if the related record in the hunting blocks lookup table has a True value in the Marked field

In the above example, the query will only count up the number of species on quota from those records in the quotas table where the matching record in the hunting blocks lookup table is equal to "True". Thus, if only the records for Luawata and Nyampala hunting blocks have a Marked value = True in the lookup table, then the query output will only include the number of species on quota from those hunting blocks.



Hunting block ID	Hunting block name	Marked
10	Lunga-Luswishi	□
11	Lupande Lower	□
12	Lupande Upper	□
13	Mulobezi	□
14	Mumbwa East	□
15	Mumbwa West	□
16	Luawata	☑
17	Nyampala	☑
18	Chifunda	□
19	Chikwa plus Fulaza	□
20	Namwala	□
21	Rufunsa	□
22	Sandwe	□
23	Sichifulo	□
24	Tendwa	□

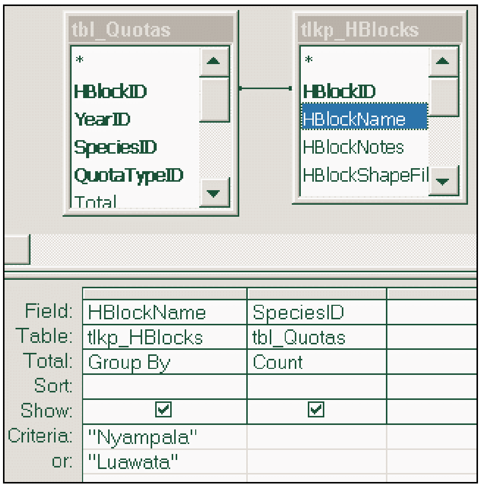
Status of the hunting blocks lookup table. Note only the records for Luawata and Nyampala have the Marked field = True (as indicated by the check mark)



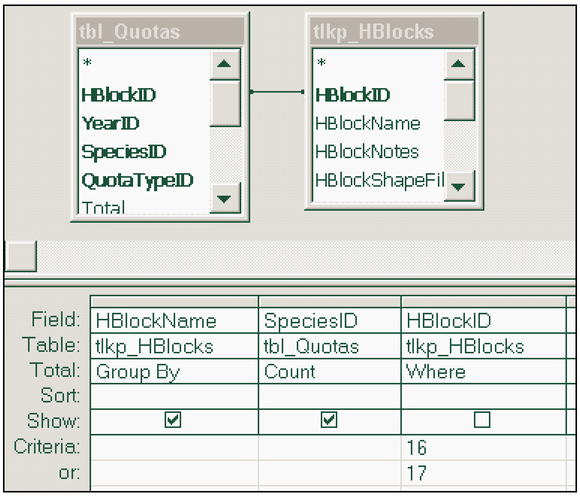
Hunting block name	CountOfSpeciesI
Luawata	71
Nyampala	82

Datasheet view of the above query. Note only those records where the matching record in the hunting blocks lookup table have a True value in the Marked field are in the datasheet. Also note that the reason why it says there are 71 species on quota is because we haven't yet set a criteria on the YearID field to limit it to just one year

There are, of course, other ways to filter a query. Instead of using the Marked field from the lookup table, we could have simply typed in the name of the hunting blocks we wish to see in the output. Alternately, you could type in the ID numbers for the hunting blocks and leave out the lookup table entirely. The following query designs will produce the exact same result as seen above:



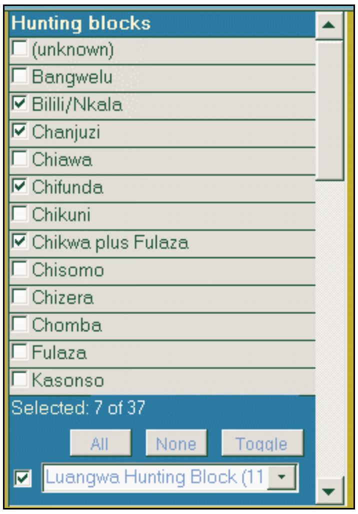
Field:	HBlockName	SpeciesID	
Table:	tlkp_HBlocks	tbl_Quotas	
Total:	Group By	Count	
Sort:			
Show:	☑	☑	
Criteria:	"Nyampala"		
or:	"Luawata"		



Field:	HBlockName	SpeciesID	HBlockID
Table:	tlkp_HBlocks	tbl_Quotas	tlkp_HBlocks
Total:	Group By	Count	Where
Sort:			
Show:	☑	☑	□
Criteria:			16
or:			17

Queries which have criteria expressions typed in manually

However the advantage of using the Marked field from the lookup to Filter queries is that we can then make use of the ADM Filter Manager. The Filter Manager, which is tightly integrated into the Menu System, displays lookup tables in a series of sub-forms and allows the user to change the value of the Marked field (i.e., check it on or off). The Filter Manager also has buttons to quickly mark or unmark all records, and can optionally provide the user a way to define groups of records in the lookup table for easy re-selection.



Hunting blocks

☐ (unknown)
☐ Bangwelu
☑ Bilili/Nkala
☑ Chanjuzi
☐ Chiawa
☑ Chifunda
☐ Chikuni
☑ Chikwa plus Fulaza
☐ Chisomo
☐ Chizera
☐ Chomba
☐ Fulaza
☐ Kasonso

Selected: 7 of 37

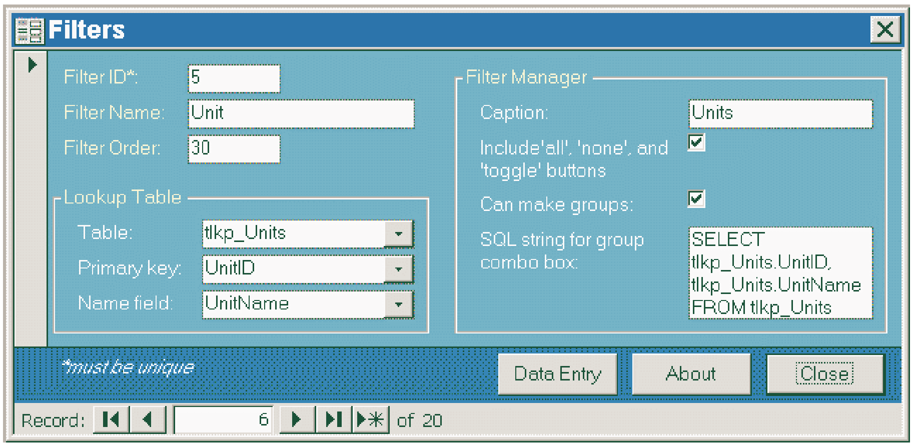
☑ Luangwa Hunting Block (11

When you filter your queries using the Marked field from the lookup table(s), the user can easily make new selections using the Filter Manager before opening the query

Thus, if you filter your queries using the Marked field from the lookup table(s), then once your object is on the Menu System users will be able to use the Filter Manager to quickly change the value of the Marked field to suit their needs. Hence if they want to see a quota list for Mumbwa East and Mumbwa West instead of Luawata and Nyamapala, they don't have retype the names of the hunting blocks, they can just select them from the list. Similarly with just a click of the mouse on the Filter Manager they can select all hunting blocks and hence produce a list of the number of species on quota for all hunting blocks.

Adding New Filters To The Filter Manager

If you have new lookup tables that you'd like to be available in the Filter Manager, then you can easily add them by selecting Filters (Admin => Maintenance) from the Main Menu.



Filters

Filter ID*: 5
Filter Name: Unit
Filter Order: 30

Lookup Table
Table: tlkp_Units
Primary key: UnitID
Name field: UnitName

Filter Manager
Caption: Units
Include'all', 'none', and 'toggle' buttons ☑
Can make groups: ☑
SQL string for group combo box: SELECT tlkp_Units.UnitID, tlkp_Units.UnitName FROM tlkp_Units

*must be unique

Each filter must have an ID number, which can be any number between 1 and 255 as long as its unique for each filter. Give your Filter a name (which will appear in the filters combo box on the Menu Manager), and an Order number (which will determine where it will appear in the filters combo box).

In the lookup table section, select the lookup table, primary key field, and a text field which describes each record. The primary key field should usually be a numeric field (or AutoNum), however any unique field will do. Note however that unless the primary key field is numeric, users won't be able to define groups of records for easy re-selection in the Filter Manager. Your lookup table must also have a Yes/No field called "Marked" in order to work with the Filter Manager. For best performance results, the Marked field should also be indexed (duplicates ok).

In the Filter Manager options section, enter a string which will appear as the caption for the filter, and check whether you want the All, None, and Toggle buttons to appear. If you want the user to be able to create groups of records for your filter, click the 'Can Make Groups' check box and type an SQL string which will be used as the record source for a two-column ID-name combo box in the define groups sub-form. If this last bit confuses you, just double-click on the SQL string box and a default SQL string will be created based on the primary key and name fields.

Once you have created a record for your lookup table, then you can start using your lookup table as a filter in the Filter Manager. You will see your lookup table listed as one of the available filters in the Menu Manager.

Adding New Objects To The Menu System

The Menu System, which includes features for documentation and filtering, is a very important tool for keeping ADM organized and usable. Chapter 2 has already explained how to use the Menu System from the users standpoint. This section describes how developers can use the Menu System to their advantage and also how to add new objects to the menu system.

Importance of Documentation
With a large database such as ADMADE's that has multiple end users and multiple architects, the importance of documentation and following consistent standards can not be emphasized enough. Without documentation of the objects used for analyses (e.g., queries), the methodology behind the analysis and will remain opaque and its repetition difficult for all but the original developer.

The Menu System offers two tools for documenting database objects. The About window provides a mechanism to enter basic information about an object, including the object's author, creation date, and purpose. This can be expounded to include explanation on the type of analysis/output the object was created for.

The Caption window provides a more dynamic description of an object, including the current status of its associated lookup tables. The primary uses of the Caption window are to (1) explain in one short paragraph the calculations and assumptions behind a graph/query/map, (2) list the selected records in the object's associated lookup tables. A typical use of a caption is to explain a graph or map for a report or presentation. Text from the caption window can be copied to the clipboard and pasted in word processor or presentation package.

Certainly not all objects will need a Caption, but it is certainly warranted for interactive graphs, maps, and other objects where the behind the scenes calculations or assumptions are not readily apparent. Text in the Caption window should be descriptive and easily understood by a layman.

Text for both the About window and Caption window can be entered when adding an object to the Menu System using the Menu Manager.

Filtering
The Filter Manager is a simple and powerful tool to build flexibility into queries, forms, and reports. With the Filter Manager, you can provide the user with a user-friendly tool to restrict the data in the output of an object by selecting items from the associated lookup tables.

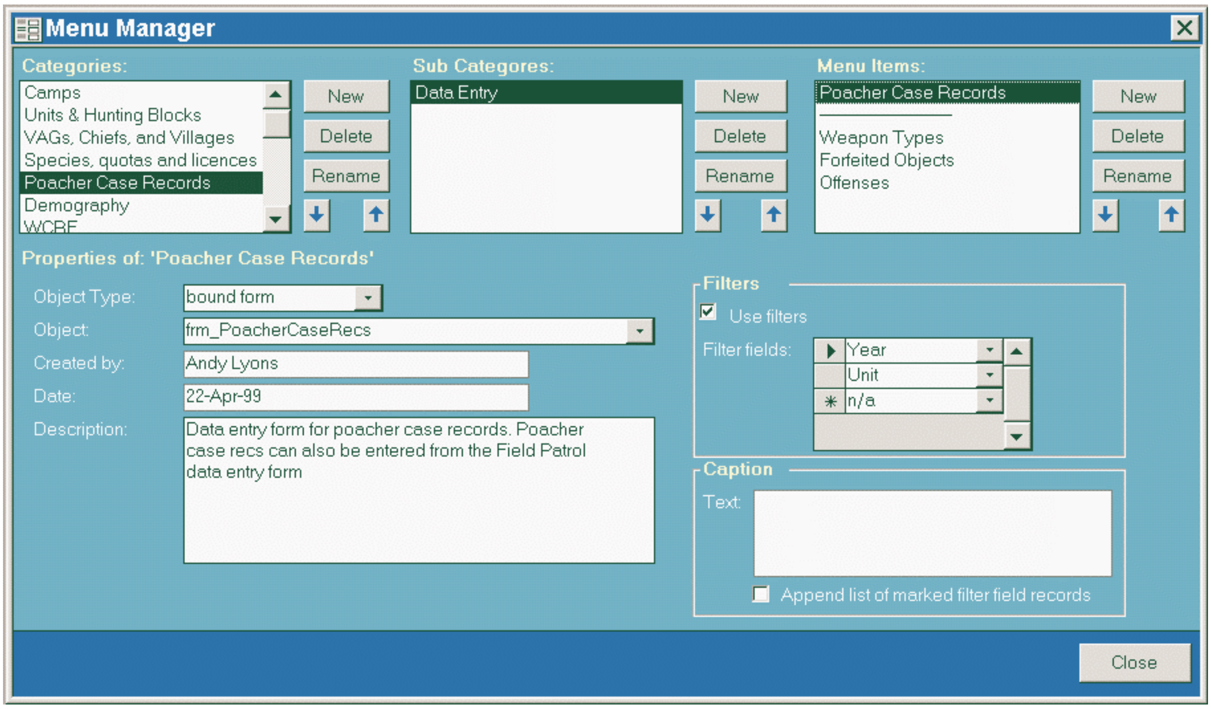
For example, suppose you create a query which calculates the average search time and trophy size of safari hunting data. By adding and joining the lookup tables for Hunting Blocks, Years, and Species to your query, and putting it on the Menu System, you now give users the flexibility to select which hunting blocks, years, and species will be included in the output. Hence whenever the user opens the query from the Main Menu, before they see the query they will see the Filter Manager where they can make their selections.

The Filter Manager can display filters for up to 6 lookup tables. Although the user can only see at most 3 at a time, if there are more than three filters a second tab will appear. Making selections in lookup tables which have quite a few records, such as the Camps lookup table, are made easier with buttons. The *All* button selects all records in the lookup table, the *None* button unselects all records, and the *Toggle* button reverses the selection. You can also define groups of lookup records. For example, you can define a group for all of the Units in the Luangwa catchment. Then whenever displaying a report or query using data for those Units, those Units can be selected with only 3 mouse clicks.

You pick the lookup table(s) which should be displayed in the Filter Manager when you add your object to the Menu System using the Menu Manager.

Menu Manager

The Menu Manager is your one-stop-shop for adding objects to the menu system. You can use the Menu Manager to add, delete, edit, or rearrange the choices available on the Main Menu. You can also create or change the menu categories or sub-categories, and enter all the information needed for your object to work with the Filter Manager, About window, and Caption window.



Properties of: 'Poacher Case Records'

Object Type: bound form

Object: frm_PoacherCaseRecs

Created by: Andy Lyons

Date: 22-Apr-99

Description: Data entry form for poacher case records. Poacher case recs can also be entered from the Field Patrol data entry form

☑ Use filters

Filter fields: Year, Unit, n/a

Text:

□ Append list of marked filter field records

The interface of the Menu Manager is fairly self-explanatory. At the top are boxes for the Categories, Sub-Categories, and Choices on the Main Menu. Next to each box are buttons to add, delete, rename, or change the order of the choices. In the Properties box you can set the properties for the current menu choice.

The text in the Description box will appear in the About window. You can also check whether you want your object to work with the Filter Manager, and select which lookup table filters should be used with your object. Obviously some objects will need no filters, however if the Use Filters box is checked the Main Menu will open the Filter Manager prior to the object. Interactive forms which require a caption can be assigned text in the Caption box. You can also set whether you want the list of selected records in the associated lookup tables to appear in the caption.

When you add an object to the Menu System, the Menu Manager may ask you if you want to hide the object in the database window. This is generally a good idea because it keeps the database window tidy and helps to prevent other users from accidentally erasing or modify your query, form, or report. Hidden objects can still be seen by checking View Hidden Objects in the Options dialog under the Tools menu. Note that the Menu Manager can only hide the primary query, form, or report. If the object has associated sub-forms, sub-reports, or sub-queries, you have to manually hide these. Objects can be hidden by selecting the object in the database Window, then selecting View—Properties and checking “Hidden”.

A feature not readily apparent on the Main Menu interface is the ability to copy and paste Sub-Categories and Menu Choices from one place to another. This may be useful as the number of objects in the database grows, and the need for different Menu Categories and Sub-Categories arises requires moving entire menu branches. To move or copy menu choices and sub-categories, right-click on the name of a menu choice or sub-category, and select copy or cut from the popup menu. Then switch to the Category or Sub-Category where you want to insert the object, right-click and select paste.

The Menu Manager can be opened from the Main Menu under the Category ‘Admin’.

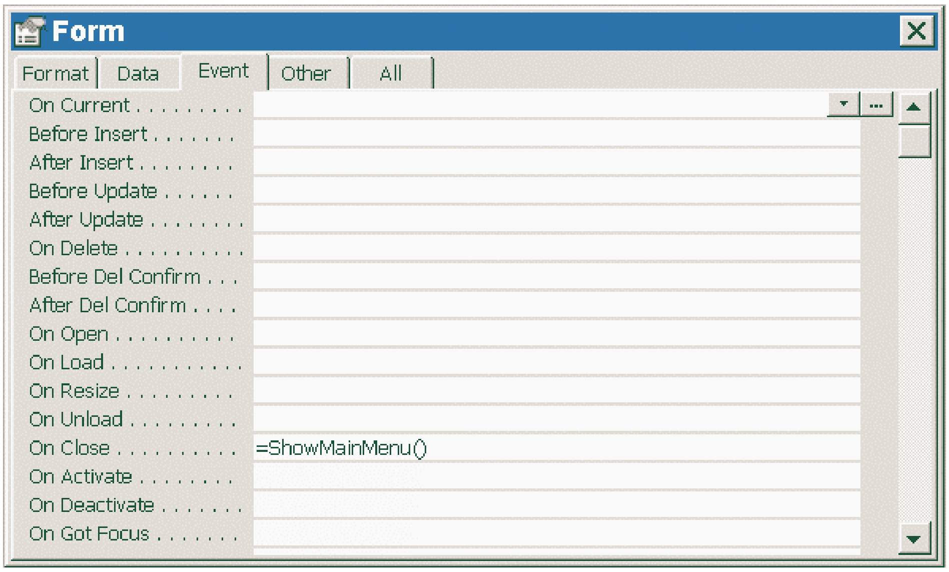
Creating a Form To Put on The Menu System

Forms which are intended to be placed on the menu system should have the certain elements, to programmatically close the form, open the about Window, or unhide the main menu. These expressions are summarized as follows:

Events which should be set on forms to work with the Menu System

Object	Event	Expression	What It Does
Close button	OnClick	=CloseMe()	Closes the form
About button	OnClick	=ShowAboutWindow()	Show the About window
Form itself	OnClose	=ShowMainMenu()	Unhides the Main Menu
Data Entry button	OnClick	=ToggleDataEntry()	Switches between data entry mode and showing all records. This is only needed for data entry forms.

To tell Access which function to run when an event occurs, enter the function name in the property window in Form design view. Make certain that you have selected the correct object (e.g., button) before you enter the expression.



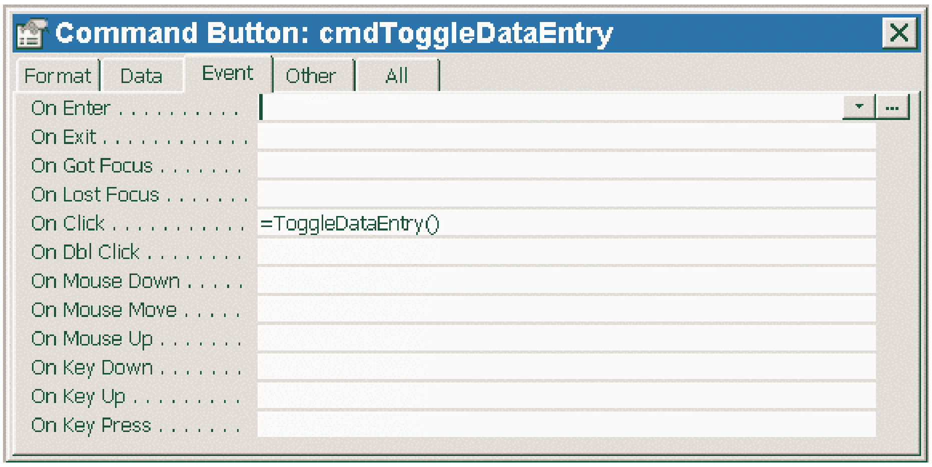
On Current . . . .
Before Insert . . . .
After Insert . . . .
Before Update . . . . . .
After Update . . . .
On Delete . . . .
Before Del Confirm . . .
After Del Confirm . . . .
On Open . . . .
On Load . . . .
On Resize . . . .
On Unload . . . .
On Close . . . . =ShowMainMenu()
On Activate . . . .
On Deactivate . . . .
On Got Focus . . . .

Also, the “Min Max Buttons” property of the form should be set to “None” so that the minimize and maximize buttons don’t appear on the form. We don’t want these buttons to appear on any forms in ADM because they’re not necessary and could result in confusion for the user because the Main Menu won’t reappear unless a form is actually closed.

Perhaps the easiest way of creating a new form with these elements is to open an existing form, save it as a new name, and then change the RecordSource to the table or query you want to base the form on.

Standard Buttons You Can Put on Your Forms

ADM has several Visual Basic functions which you can easily attached to command buttons to perform everyday operations, such as closing the form, showing the main menu, etc. To attach code to a command button, you simply enter the name of the function in the property window for that button:



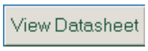
On Enter . . . .
On Exit . . . .
On Got Focus . . . .
On Lost Focus . . . .
On Click . . . . =ToggleDataEntry()
On Dbl Click . . . .
On Mouse Down . . . . .
On Mouse Move . . . . .
On Mouse Up . . . .
On Key Down . . . .
On Key Up . . . .
On Key Press . . . .

Adding a Visual Basic function to a command button

The available VBA functions and their corresponding buttons are summarized in the table below:

Button	OnClick expression	Description
Close	=CloseMe()	Closes the active form

	=ShowAboutWindow()	Opens the About window
	=ToggleDataEntry()	Switches to data entry mode for the current form. In data entry mode, all existing records are hidden and the user is presented with a new blank record. If data entry mode is already enabled, this function will switch back to showing all records. Note that the Caption (i.e., text) for a command button using this function *must* be ‘Data Entry’.
	=ShowCaptionWindow()	This button will open the caption window, which describes the current form and lists the values in the lookup table(s) which are currently including in the objects recordset. Of course this requires that the author has entered a caption for the form (using the Menu Manager). Captions are usually only needed for interactive forms, such as interactive graphs and maps, where it isn’t obvious what records are being shown.
	=FormFilters("Graph")	This button will open up the Filter Manager so the user make new selections. If you pass the name of a control as an argument to the function (i.e., type the name of a control in the parenthesis) then the function will refresh the control after the user has closed the Filter Manager. For example, if the form contains an MS Graph control named ‘Graph’, then after the Filter Manager is closed the graph will be refreshed to reflect the changes. See also Creating an Interactive Graph. If you pass an empty string to the function (i.e., “”), then the function will requery the form. This is useful if the record source for the form itself uses filters from the lookup tables (e.g., see the Data Log form). Using the Filters button to open the Filter Manager of course requires that the author has already entered (in the Menu Manager) the names of the lookup table(s) which are joined to the graph query or form record source. Note: this is **not** the Visual Basic function you should use to open the Filter Manager on an interactive map. See Using Dynamic Maps on Forms and Reports.
	=ShowHelp(10)	This function will open the ADM Online Users Guide (a Windows help file). The number which is passed to the function is the context ID number for the Help topic you want to see. 10 is the number for the main table of contents topic. You can enter context ID numbers for other topics as you write the Help file.

	=ShowGraphDatasheet("Graph")	This function will create and open a new query which contains the data for an MS Graph on the form. The parameter passed to the function must be the name of an MS Graph control on the form. This button is only useful for those forms which have graph controls on them. Note that once the query with the data for the graph is opened, the user must still save it manually or it will be lost.

Another very important Visual Basic function which you can use on your forms and reports is =ShowMainMenu(). This function should be entered for the OnClose property of your forms and reports, so that when the user closes the form or report the Main Menu will become unhidden.

Adding Data Logging to a Data Entry Form

You can easily add Data Logging capability to your data entry forms to keep track of when users make changes to the tables using your form. The steps for doing this are basically:

1. Enter the name of the table your form is bound to
2. Enter the name of your form, and specify which controls contain the information which you want to use to identify records
3. Enter the appropriate Visual Basic function names in the Form property sheet

1. Enter the name of the table your form is bound to

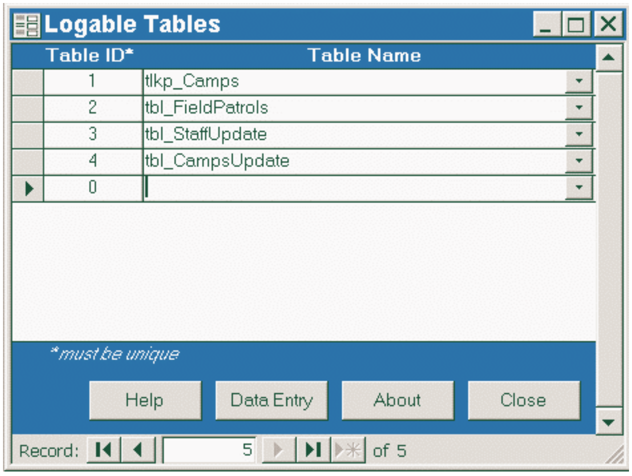


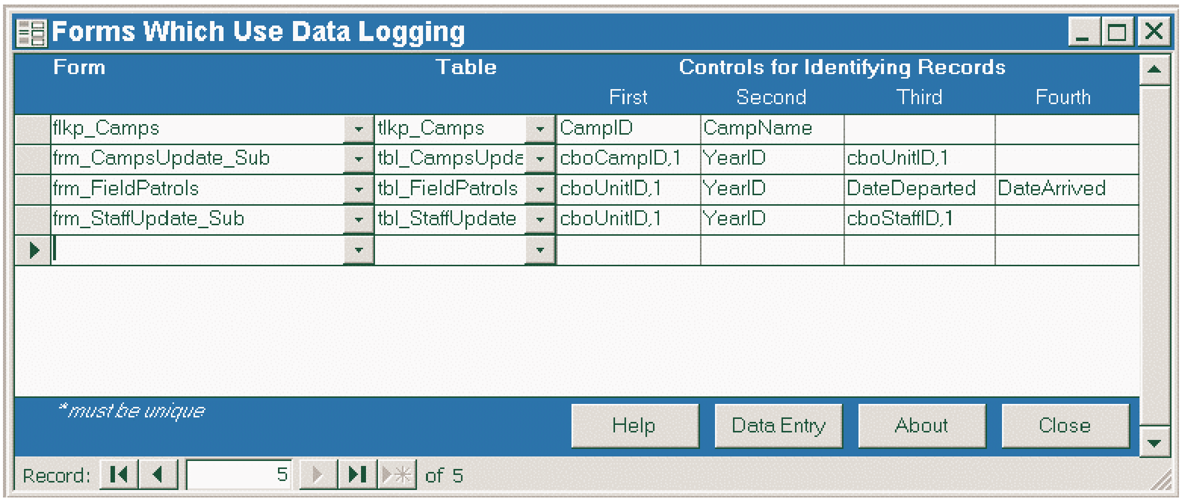
Table ID*	Table Name
1	tlkp_Camps
2	tbl_FieldPatrols
3	tbl_StaffUpdate
4	tbl_CampsUpdate
0	

In Logable Tables, (from the Main Menu select Data Logging, then Setup) enter a unique ID number for the table which your data entry form is bound to.

Note: if your data entry form has sub-forms for detail records, then enter the name of the parent table. For example, if your form is a data entry form for safari hunting data, then you would make a record for tbl_Hunts (the parent table), not tbl_HuntsTrophies (a details table).

2. Enter the name of your form, and specify which controls contain the information which you want to use to identify records

In Logable Forms, (from the Main Menu select Data Logging, then Setup) enter information about your data entry form. The Visual Basic code will use this information to know how to make the appropriate records in the log table.



Forms Which Use Data Logging

Form	Table	Controls for Identifying Records			
		First	Second	Third	Fourth
flkp_Camps	tlkp_Camps	CampID	CampName		
frm_CampsUpdate_Sub	tbl_CampsUpde	cboCampID,1	YearID	cboUnitID,1	
frm_FieldPatrols	tbl_FieldPatrols	cboUnitID,1	YearID	DateDeparted	DateArrived
frm_StaffUpdate_Sub	tbl_StaffUpdate	cboUnitID,1	YearID	cboStaffID,1	

*must be unique

Record: 5 of 5

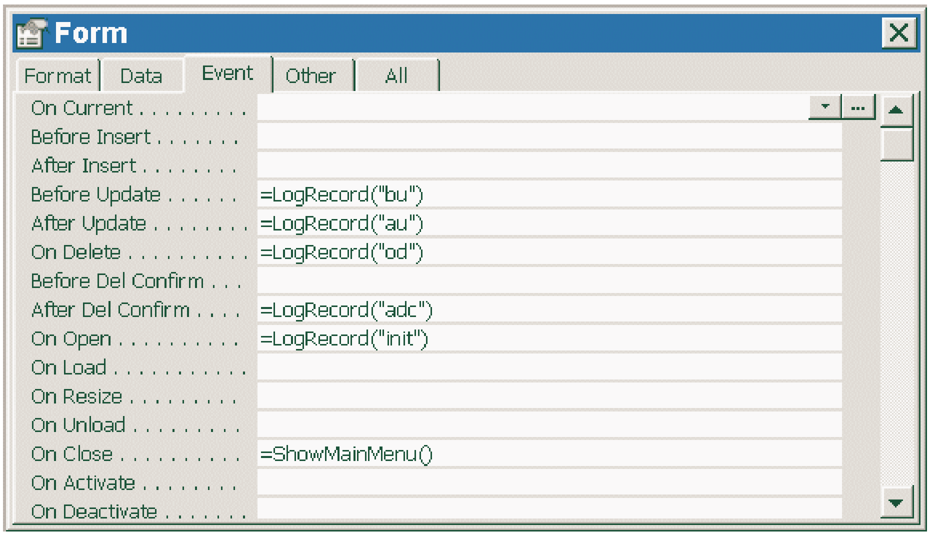
In the first two columns, enter the name of your form and the table it is bound to. Then enter the name of up to four controls on your form which together will uniquely identify a record. You don’t have to enter all four controls, but you should have at least one. For example, if you setting up a data entry form for safari hunts, you might want to include the name of the client, the PH, and the beginning and ending dates of the hunt. The records in the log table will contain this information, so you can easily go back to the record or retrieve the data form if problems arise.

Note: it is generally not a good idea to use a numeric primary key field as the record identifier which will be stored in the log. For example, even though each hunt in tbl_Hunts has a unique ID number, when you see these numbers printed out on the Data Log report you won’t really know which hunt it was. And if the hunts table should become corrupted and have to be recreated or restored from a backup, those ID numbers will be useless. It’s better to use controls which contain text information, such as VAGName, Employee Name, etc., to identify records in the log table.

If the control you want to use is a combo or list box, then you should also follow the name of the control with a comma and the number of the column you want to use. Note that in Access the index number for the first column of a combo box is 0, the second column is 1, and so on. For example, most combo boxes have two columns, the first column (which is usually hidden) has a numeric value, and the second column (which the user sees) has text. If you want to identify the current record in the data log using information from the combo, you’d be better off using the second column which contains text. So if, for example, your data entry form has a combo box control named cboStaffID which contains a hidden numeric field followed by the person’s last name, you would enter: cboStaffID,1.

3. Enter the appropriate Visual Basic function names in the Form property sheet

The last (and easiest) step to adding Data Logging capability to your data entry for is to enter the following Visual Basic functions on the form property sheet:



On Current	
Before Insert	
After Insert	
Before Update	=LogRecord("bu")
After Update	=LogRecord("au")
On Delete	=LogRecord("od")
Before Del Confirm . . .	
After Del Confirm	=LogRecord("adc")
On Open	=LogRecord("init")
On Load	
On Resize	
On Unload	
On Close	=ShowMainMenu()
On Activate	
On Deactivate	

Note the function names should be typed exactly as shown above, including the function arguments (i.e., the text inside the parenthesis). Note the ShowMainMenu() function in the property sheet above is for the menu system, it has nothing to do with data logging.

These functions will be called when the user changes, adds, or edits a record. Note however that if the user uses other ways to make changes to the tables (for example by running a query, pasting in records from the clipboard, or changing records programmatically) then these functions will not be run. In these cases, you have to use other Visual Basic code to trap the changes.

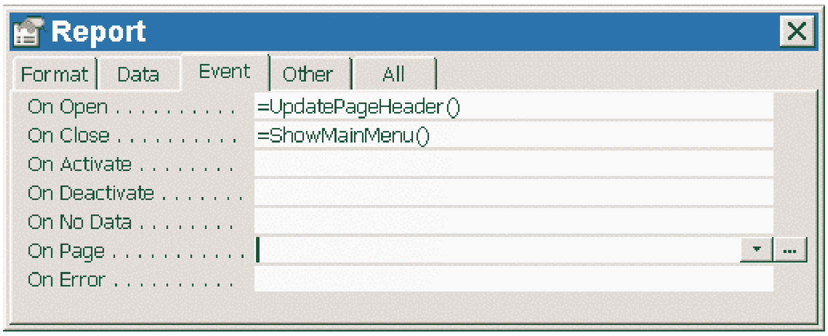
Adding a Report to the Menu System

Adding a report to the Menu System is very similar to adding a query or form. The basic steps to follow are:

1. If you want the user to be able to filter the records in your report using the Filter Manager, make certain that (1) the relevant lookup tables are part of the query/SQL which the report is based on, and (2) the Marked field in each of the lookup tables appear in the query grid with ‘True’ as the criteria expression.

2. Add the report to the Menu System using the Menu Manager. Note that reports have more field to fill in on the Menu Manager than forms or queries: a default page header. The text you enter as default page header will appear each time the user opens the report, but can be over-ridden in case the user wants the page header to be customized.

3. In order for the Custom Page Header to work, your report should have a label in the PageHeader section named lblPageHeader, and the following function name in the report OnOpen property:



On Open	=UpdatePageHeader()
On Close	=ShowMainMenu()
On Activate	
On Deactivate	
On No Data	
On Page	
On Error	

Also note that in the property window above the OnClose expression is =ShowMainMenu(), so that when the user closes the report in Print Preview the Main Menu will be unhidden.

Incorporating Picture/Graphic Data in ADM

If you'd like to include picture data into a dataset (e.g., drawings of animals, photographs of community development projects, images of staff) there are two strategies you can use. You can either save the picture data directly into a table (as an OLE field), or you can keep the image as a separate file on disk and just save the path and file name in a table (as a text field). There are advantages and disadvantages of both strategies:

Saving Images in an Access Table as an OLE Field

<u>Advantages</u>

- It's easy to transport the pictures from one place to another (just copy the MDB file)
- Its easy to make forms/reports that display the image(s)
- Images get backed up whenever you backup the MDB file

<u>Disadvantages</u>

- Access does not save images in compressed format, so your MDB file will become very large very quickly
- It's a little awkward getting the images into the table (must paste from clipboard, use the insert object command, or develop your own VBA code)

Saving Images as Separate Graphic Files and Storing only the Filename in an Access table

<u>Advantages</u>

- the images can be used by other programs (such as MS Word)
- the images can be saved in a compressed format, such as JPG or GIF, thus significantly reducing consumed file space on the hard drive
- The database will only save a short text string (the file name), so your MDB file won't become bloated
- You can still use the images in Access forms and reports

<u>Disadvantages</u>

- If you install ADM on another machine, you have to make certain that the image files also get copied and fall into the same directory structure.
- You have to write VBA code to display the images in forms or reports
- Whenever you want to add or delete an image, you have to add/delete it in two places: (1) the image itself must be in the appropriate directory on the hard drive, (2) you have to enter the path/filename to the image in Access
- Images have to be backed up separately from the MDB file

In general, if you expect to use a large number of images, or any number of large images (a 640x480 photograph from a digital camera should be considered a large image), then you're probably better off saving the files externally. Your MDB file won't become bloated which will improve data access speed for all tables and reduce size of back up files.

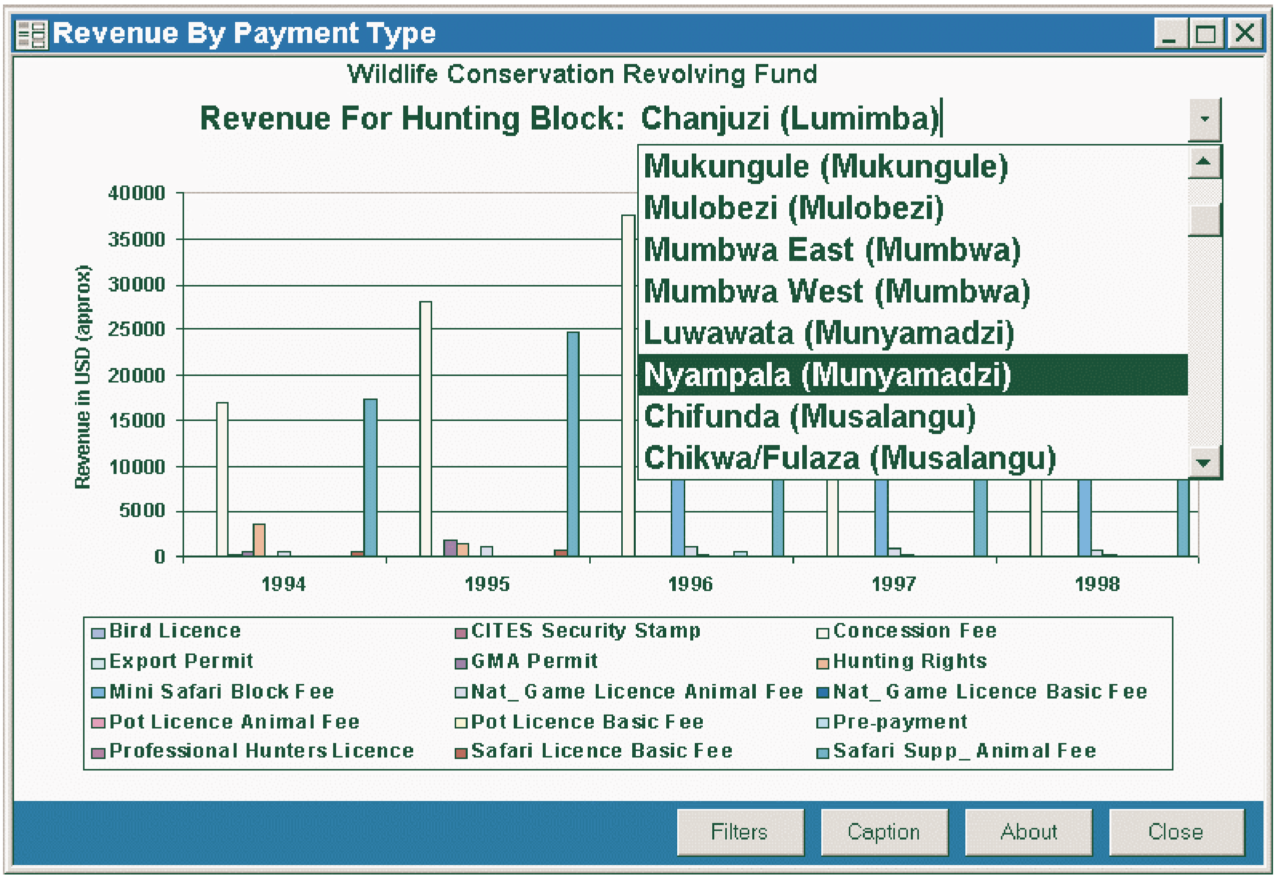
To display images whose file names are stored in an Access table, you need to use a short VBA routine for the OnCurrent event (for a form) or the OnFormat event (for a report). Suppose you have a picture control named ‘Img1’ on your form, and a field called ImageFile in your table. Then the code that should go with the OnCurrent property of a form is:

```
strPathToImages = GetADMSetting("PathToSpeciesImages")
Me!Img1.Picture = strPathToImages & Me!ImageFile
```

The first line of code here gets the PathToImages value from the GetADMSetting function, and assigns it to a string variable. The second line code sets the Picture property of the Img1 control to the value of the ImageFile field.

Interactive Charts

Interactive charts and graphs are powerful tools for visualizing your data. Graphs created by MS Graph can be customized by the user, and/or copied and pasted to other programs such as MS Word. By using the Filter Manager or linking your chart to a combo box on the form, you can make your chart flexible and interactive, so it will be ready for a variety of purposes.



Revenue For Hunting Block: Chanjuzi (Lumimba)

Access makes it easy to insert graphs onto forms and reports, which can then bound to an underlying record sets or other controls on the form. The general elements on a form containing an interactive graph will typically include:

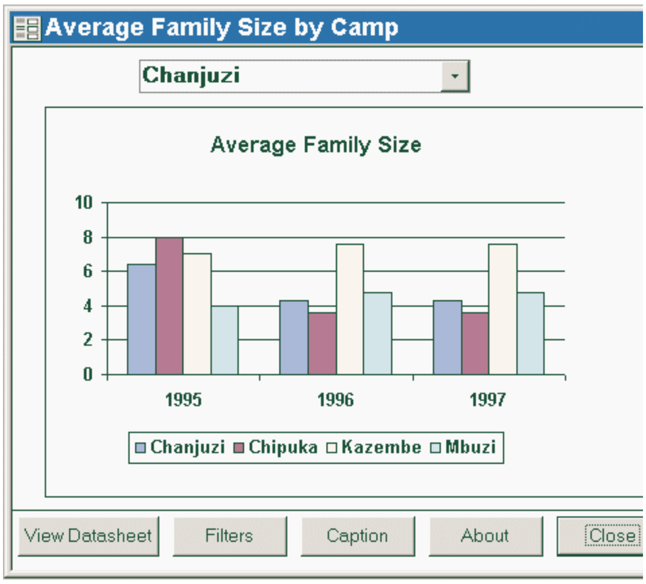
- an MS Graph control whose row source will typically be a query which includes the Marked Yes/No field from one or more lookup tables (so that the user can use the Filter Manager to select which records should be included in the graph). The MS Graph control should be enabled (so users can customize the appearance of the graph), but locked (so user changes are not saved).

- a Filter button which will open up the Filter Manager and refresh the graph control
- a Caption button which will display text describing the current state of the graph in the Caption window.
- a View Datasheet button which will create and open a query showing the data the graph is based on
- optionally, an unbound combo or list box displaying one of the filter fields for the graph. The AfterUpdate event of the combo box changes the RowSource of the graph control to reflect the users new selection in the combo box

The best way to demonstrate how these different elements can be combined to create an interactive graph is to walk through an example.

Exercise: Creating an Interactive Graph

Creating interactive graphs is a piece of cake in Access. In this exercise, we will create a bar chart which shows the average family size at each camp in a Unit for the selected years. The end product will look something like the chart on the right. Users can select the Unit they wish to see in the combo box at the top of the form, and the graph will show the average family size for the employees of each camp. The use can also click the Filters button to open the Filter Manager whereby they can select which years and types of employees (e.g., which position and status) to include in the graph.



The steps in creating this type of graph are:

1. Create an unbound form with a graph control, buttons, and any other controls that are needed to filter the graph
2. Create a query which will be used as the RecordSource for the graph
3. Using Visual Basic, set the RowSource property of the graph control on your form to the SQL string of your query. Link any query parameters in the SQL statement to controls on the form.
4. Add the form to the menu system.
5. Customize the display of the graph.

These steps are described in more detail below.

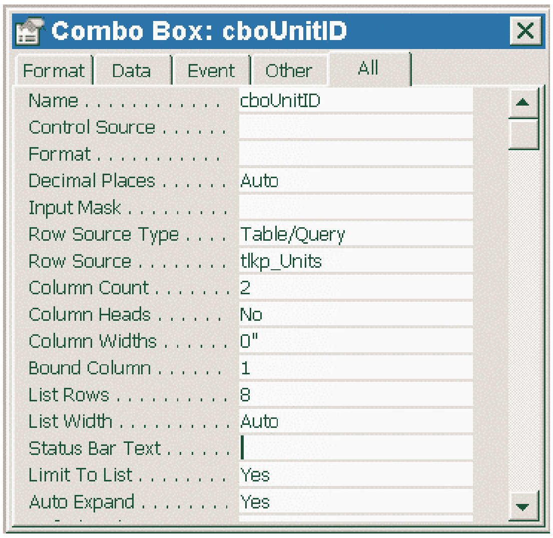
1. Create an unbound form with a graph control and any other controls that are needed to filter the graph

The first step is to create a form which will be used to show the graph; the graph will be embedded on the form as a graph control. The form should be unbound, meaning it isn’t linked to any table (the graph on the form however will be have its RowSource property set to an SQL string). The form should also have the following controls and properties:

Properties
ScrollBars – neither
NavigationButtons – no
RecordSelectors – no
Caption – Average Family Size By Camp
OnClose – =ShowMainMenu()

Controls
About button, OnClick: =ShowAboutWindow()
Close button, OnClick: =CloseMe()
Filters button, OnClick: =FormFilters("Graph")
Caption button, OnClick: =ShowCaptionWindow()
cboUnitID combo box, AfterUpdate: =UpdateGraph()
Graph – graph control

For the UnitID combo box, we want the value of the combo box to return the UnitID (a numeric value) of the selected Unit, but the user should see UnitName. Hence we need a two column combo box with the row source from the Units lookup table, where the first column is the UnitID and the second column is the UnitName. The combo box should be bound to column 1, meaning that the combo box gets its value from column 1 or the ID number, and the width of column 1 should be set to 0 (so the user doesn't see it). Also make certain that the name of the combo box is cboUnitID, because that's the name we will use later when we link the RowSource property of the graph control to the combo box.

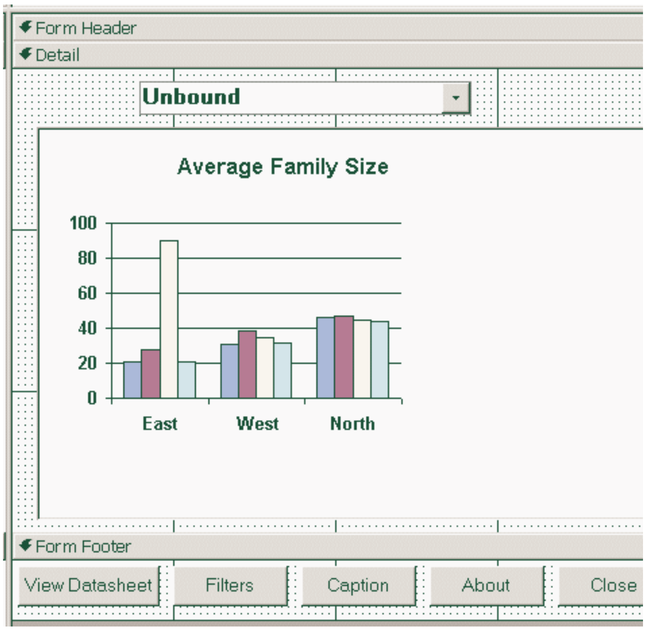


Name	cboUnitID
Control Source	
Format	
Decimal Places	Auto
Input Mask	
Row Source Type	Table/Query
Row Source	tlkp_Units
Column Count	2
Column Heads	No
Column Widths	0"
Bound Column	1
List Rows	8
List Width	Auto
Status Bar Text	
Limit To List	Yes
Auto Expand	Yes

The property sheet for the combo box should look like the one at the right.

To add a graph control to your form, select Chart from the Insert menu and drag a box on the form where you want the chart to go. Usually MS Access starts the Chart Wizard when you do this, but just click the *Finished* button at the first wizard dialog because we are going to customize the graph ourselves. Be sure to change the name of the graph control to "Graph". Also, you need to set the enabled property of the graph control to yes, and the locked property of the graph to no, so that later we can modify how the graph displays our data.

With all these elements in place, our form now looks something like the one on the right. Don't worry that the graph is showing some bogus data, we haven't linked it to our query yet or changed the display settings.



2. Create a query which will be used as the RecordSource for the graph

To help you make a query which the graph will use as a record source, think about how you would create this type of graph in a spreadsheet program like Excel. In this example, to create a graph like the one above, we need a query which can output the following fields:

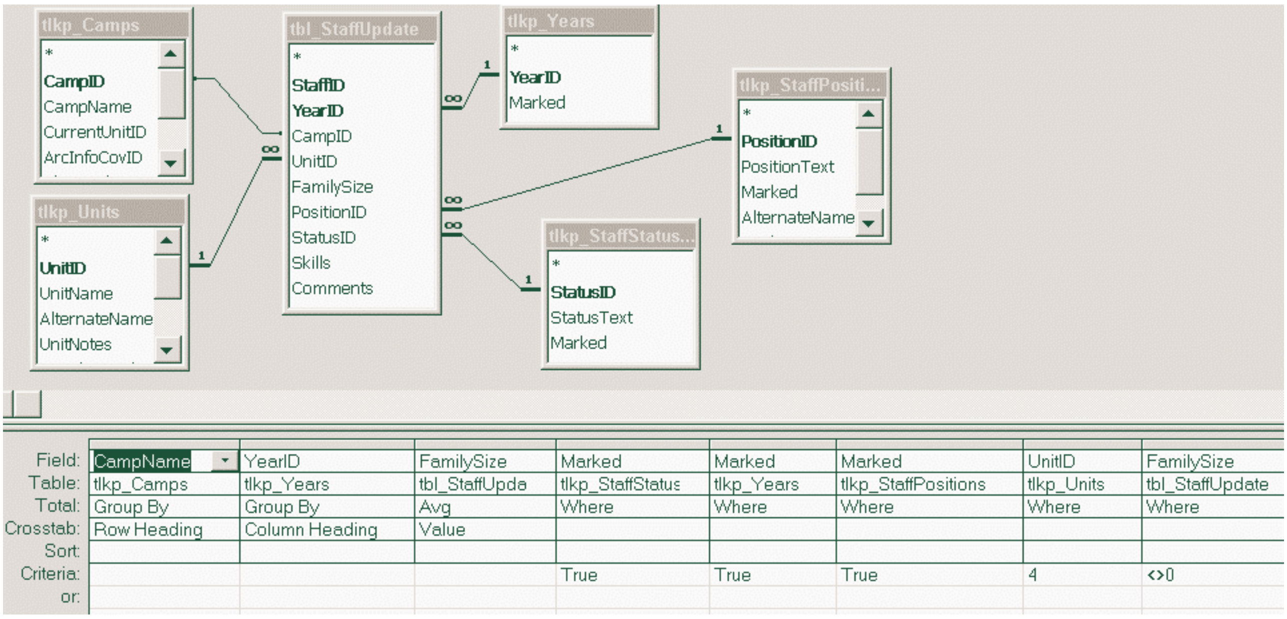
CampName	1995	1996	1997	1998
Leulo	x	x	x	x
Mansa	x	x	x	x

We know that if we can make a query which produces the fields above, MS Graph will be able to make the kind of bar chart we want. Notice that the name of the Unit is not included as one of the fields in the query. That is of course because MS Graph doesn't care or need to know which Unit is being graphed to produce the bars. The UnitID will be used to filter the recordset, but will not be included as one of the fields of the graph RowSource itself.

After you know what fields should be in your query, think about how it will be filtered. There are two types of mechanisms that can be used to filter a query. You can filter the records in the query by joining lookup tables to the query and setting criteria for the Marked field in the lookup tables (i.e., Marked field = True). This will restrict the query to include only those records where the matching records in all the lookup tables has a Marked value of True (which can be easily set by the user using the Filter Manager). You can also design a query so that the output is filtered based on a numeric value, such as the UnitID, which is linked to a control on the form.

In our example, we are going to use both methods of filtering. The data for our graph is going to come primarily from the tbl_StaffUpdates table, which has fields for year, family size, camp and Unit. The field that the user will want to change most frequently when interacting with the graph is the Unit, so we will link the RowSource property of our graph to a combo box on the form which shows the Units. We will also give the user the option (using the Filter Manager) to restrict the graph based on matching records from the lookup tables for staff position, status, and year. That way, the user can be very selective, so for example, they could specify that the graph should only show the average family size for village scouts who are in camp for the years 96-97.

Based on how we have decided that we want our interactive graph to function, the query for the graph will look like this:



Field:	CampName	YearID	FamilySize	Marked	Marked	Marked	UnitID	FamilySize
Table:	tlkp_Camps	tlkp_Years	tbl_StaffUpda	tlkp_StaffStatus	tlkp_Years	tlkp_StaffPositions	tlkp_Units	tbl_StaffUpdate
Total:	Group By	Group By	Avg	Where	Where	Where	Where	Where
Crosstab:	Row Heading	Column Heading	Value					
Sort:								
Criteria:				True	True	True	4	<>0
or:								

Note that our query is a crosstab query that uses the CampName as the row heading, YearID as the column headings, and average family size as the values. Also note that there are three lookup tables joined in the query, Units, StaffPositions, and StaffStatus, and that the Marked field for these tables has a criteria requirement set to True. Hence only those records from tbl_StaffUpdate where the matching records in all three lookup tables have Marked = True will be used in the output.

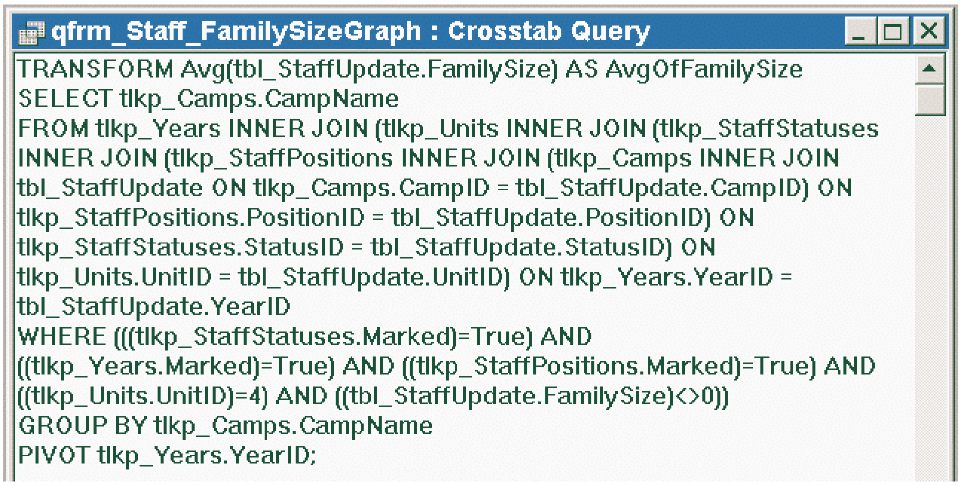
Also note that there is a criteria expression for UnitID. However unlike the other lookup tables, the criteria for UnitID is a number. Right now the UnitID is arbitrarily set to ‘4’, the UnitID for Chanjuzi. But later we’ll link this parameter to the UnitID combo box on the form using Visual Basic so the user can select the Unit interactively.

Finally, the last column in the query by design grid sets a criteria for the family size, so that only those records in tbl_StaffUpdates where the family size is not zero will be averaged.

Once you have created the query, we’re ready to put it on the graph. But we won’t actually use the query itself, but its SQL statement, a specially formatted string defining the query which Access create behind the scenes.

<u>3. Using Visual Basic, set the RowSource property of the graph control on your form to the SQL string of your query. Link any query parameters in the SQL statement to controls on the form.</u>

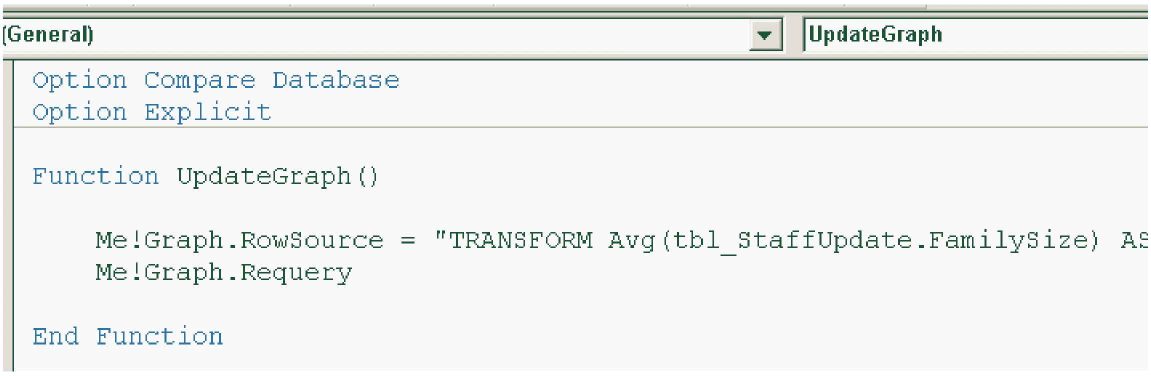
Select View…SQL from your query and copy the entire SQL statement to the clipboard. The SQL statement is the string which Access creates behind the scenes and defines your query.



```
TRANSFORM Avg(tbl_StaffUpdate.FamilySize) AS AvgOfFamilySize
SELECT tlkp_Camps.CampName
FROM tlkp_Years INNER JOIN (tlkp_Units INNER JOIN (tlkp_StaffStatuses
INNER JOIN (tlkp_StaffPositions INNER JOIN (tlkp_Camps INNER JOIN
tbl_StaffUpdate ON tlkp_Camps.CampID = tbl_StaffUpdate.CampID) ON
tlkp_StaffPositions.PositionID = tbl_StaffUpdate.PositionID) ON
tlkp_StaffStatuses.StatusID = tbl_StaffUpdate.StatusID) ON
tlkp_Units.UnitID = tbl_StaffUpdate.UnitID) ON tlkp_Years.YearID =
tbl_StaffUpdate.YearID
WHERE (((tlkp_StaffStatuses.Marked)=True) AND
((tlkp_Years.Marked)=True) AND ((tlkp_StaffPositions.Marked)=True) AND
((tlkp_Units.UnitID)=4) AND ((tbl_StaffUpdate.FamilySize)<>0))
GROUP BY tlkp_Camps.CampName
PIVOT tlkp_Years.YearID;
```

When you create a query in Access, behind the scenes it is making an SQL statement, a specially constructed and universally understood phrase which defines the query. We will use this string as the RowSource property on our graph control, and then manipulate it with Visual Basic so that it is linked to the UnitID combo box.

Next switch to your form, and select View…Code. This will show you the Visual Basic module which is saved with the form. Insert a new function called UpdateGraph by selecting Insert...Function from the menu. When you’re done, the function should have only two lines as shown:



```
Option Compare Database
Option Explicit

Function UpdateGraph()

    Me!Graph.RowSource = "TRANSFORM Avg(tbl_StaffUpdate.FamilySize) AS
    Me!Graph.Requery

End Function
```

As shown above, the RowSource property of the Graph control should equal the SQL statement of your query, which you paste in from the clipboard. Normally when you paste in an SQL string from the clipboard, it is split into several lines, which you need to correct manually. Don't worry that the SQL string runs beyond the right side of the screen.

Next, we need to modify the SQL string so that the UnitID parameter specified in the SQL is linked to the UnitID combo box. Scroll to the left until you see the part of the SQL string that says:

```
... (tlkp_Units.UnitID) = 4) ...
```

Modify the string as follows so that the value of the combo box is used as the criteria for the UnitID field, instead of the number 4 which we were just using temporarily:

```
... (tlkp_Units.UnitID) = " & Me!cboUnitID & ") ...
```

Now, whenever that function is called, the RowSource property for the graph will be updated and the graph will be re-queried to show data for the selected unit. Because you already set the AfterUpdate property of the UnitID combo box to =UpdateGraph() when you created the form, whenever the user changes the value in the combo box, the AfterUpdate event will call the UpdateGraph() function and the graph will be requeried. Try it and see. Go to form view and select a Unit which has records in the tbl_StaffUpdates. If you don't see any data, then perhaps some of the filters fields are not marked (e.g., year, staff positions, and staff status). After you put the form on the Main Menu system, you can click the Filters button and see.

4. Add the form to the menu system

Add your new form to the menu system using the Menu Manager. For filters, select Year, Position, and Staff Status. Enter a meaningful caption, such as "This graph represents the average family size for each camp for the following types of staff positions and status:". If you check the 'Append filter fields" box, then ADM will list the selected records in the positions, years, and staff statuses lookup tables.

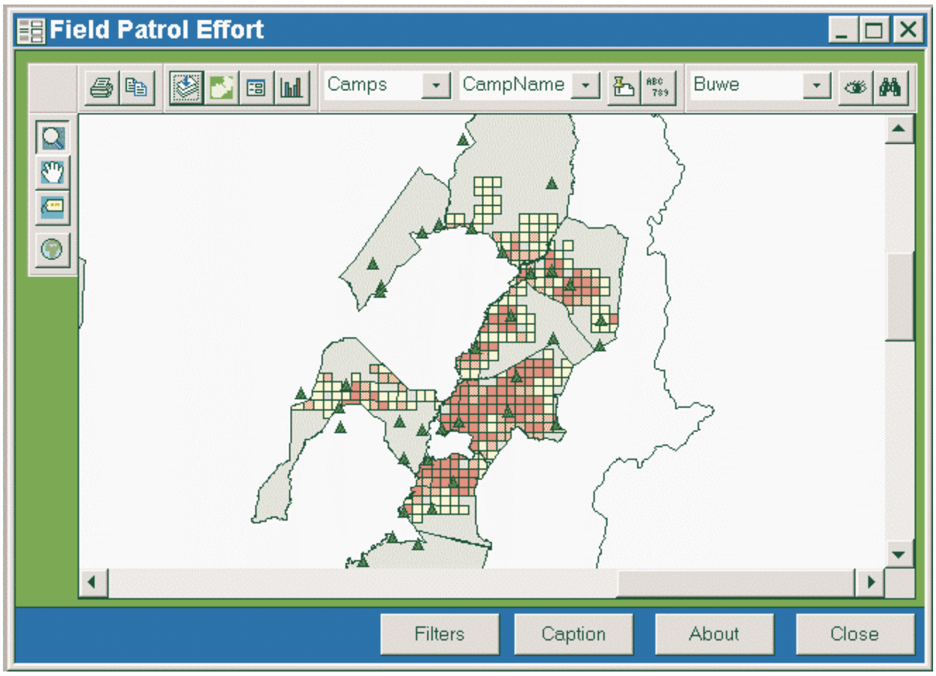
Once you add the graph to the menu system, the About, Caption, and Filters buttons should all work. Test them and see.

5. Customize the graph

If you did everything correct, your graph should now be linked to the combo box on the form, as well as the lookup tables. However the graph is probably not showing the data the way you want because we haven't changed the default chart options. In Form view, double-click on the graph and change the format of the chart elements as you like. For example, you can make it a line graph, bar chart, column chart, 2D or 3D, pick the colors etc. When you're all done, go back to design view, and set the Locked property of the graph to Yes. This way, users can still modify the graph if they'd like to suit their particular purpose, but their changes won't be saved with the form. They can also copy and paste the graph into other Windows programs like PowerPoint.

Interactive Maps

ADM's built-in mapping capabilities allow users to visualize tabular data spatially. While these features are not a substitute for a full-fledged GIS package such as ArcView, they do provide highly integrated and dynamic displays of monitoring data through maps.



Creating new maps for use in ADM is not difficult once you understand a few basic concepts. This opens up exciting possibilities for users who want to visualize the spatial relationships of monitoring data.

A Basic Primer on How ADM Creates Maps

- Maps are used in three different settings in ADM:
  - Interactive Maps are intended to be viewed primarily on-screen, although they can also be printed, copied to the clipboard, or sent to PowerPoint. The toolbar on Interactive Maps is standard and offers a variety of buttons to allow a user to customize the display of data.
  - Maps can also be included on a Slide Template. Slide Templates are powerful tools which allow you to automatically create new visual presentations of data using PowerPoint
  - Some specialized ADM forms, such as the Unit Profile form and Handy Reference Map, contain maps which are programmed to perform certain functions particular to that form.
- Interactive Maps and maps for Slide Templates can be created without any programming knowledge. Creating specialized map forms may require some programming experience.
- Regardless of how they are used, all types of maps in ADM are based on a Map Definition, which defines the layers (e.g., themes) which will appear in the map and how those layers should be displayed.
- All maps in ADM are drawn using one or more shape file coverages. Each shape file represents a certain type of feature (e.g., rivers, hunting block boundaries, camps, etc.). Unlike ADMADE's former GIS, the shape files used in ADM are national coverages, meaning they contain features for the entire country as opposed to just one GMA. The shape file covers should be saved in a common directory (specified in Program Options).
- There are two shape file covers for the 5 km^2 grids. One grid layer is for field patrols, and the other is for hunting data (the two main datasets which are geo-referenced by grid). This is necessary because in some areas different base maps are used for field patrols and safari

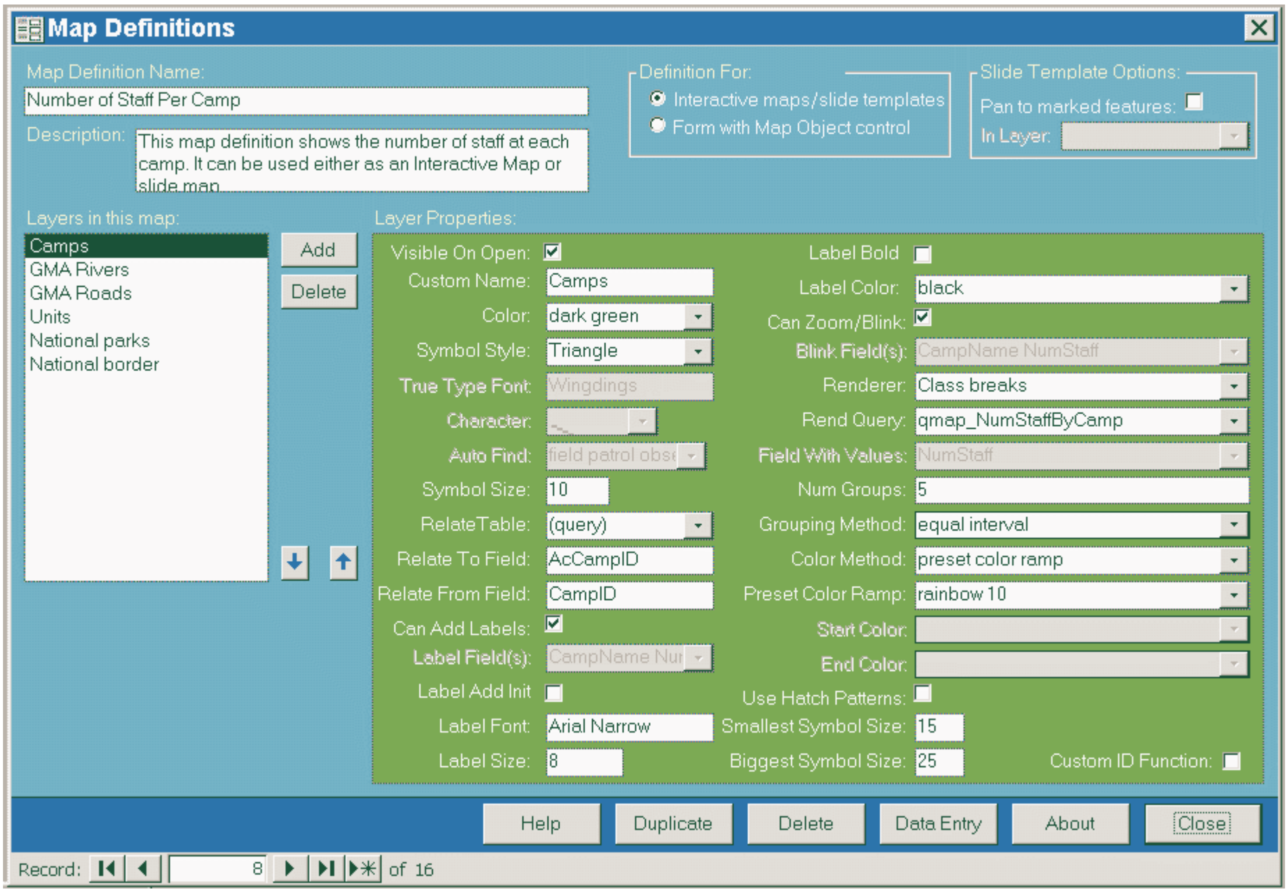
monitoring. In addition, there is also a polygon grid cover (for displaying grid data using colors) and a point grid cover (for displaying grid data with graduated symbols).

- The attribute table of a shape file can be joined to a table or queries in Access, so that you can add labels or classify features in the shape file based on name fields or attributes from tables or queries in Access.
- Shape files are usually joined to Access tables based on a common key field which is usually the primary key of one of the Access lookup tables. In other words, most shape file attribute tables used to displaying data have a key field used just for creating joining to ADM tables. Each feature in the shape file has a value in this key field which matches the primary key of one of the records in a lookup table in Access.
- There are four possible options for rendering (i.e., coloring) features in a layer. These include class breaks (where the color of the feature represents the value of some field, such as income), graduated symbols, pie charts, and bar charts. *NOTE: Pie charts and bar charts are not currently working due to a bug with MapObject and Access.* MapObjects supports other classifying options such as dot density, but these have not been setup yet to work in ADM.
- All of the programming required by Interactive Maps and Slide Templates (e.g., procedures to zoom in and out, blink selected features, etc.) is contained in a standard module. In other words, all ADM maps use the same Visual Basic functions to load layers, classify them, and respond to tool bar button clicks. This makes it easy to create new maps without any programming knowledge by simply using this same ‘template’.
- ADM uses Map Objects to programmatically create maps based on Map Definitions. Map Objects is an ActiveX control produced by ESRI which among other capabilities allows the programmatic integration of tabular data and GIS covers

Creating a Map Definition

Creating a new map definition is relatively easy in ADM. Before you go to the Map Definition form, you should think how you want your map to look and function. Will it be used as an Interactive Map, or on a Slide Template? Which layers should have labels? Will any of the layers be joined to a query? Which layers do you want visible when the map first opens, and which do you want to be available when the user clicks on the ‘Other Layers’ button? Which layers should be joined to an Access table or query so that you can classify them and/or label the features? Which ones should be on ‘top’ of the others?

Once you have determined how your map will function and look, open the Map Definitions form from the Main Menu, by selecting Admin => Mapping Setup => Map Definitions


Map Definitions

Map Definition Name: Number of Staff Per Camp

Description: This map definition shows the number of staff at each camp. It can be used either as an Interactive Map or slide map

Definition For: ● Interactive maps/slide templates ○ Form with Map Object control

Slide Template Options: Pan to marked features: □ In Layer:

Layers in this map:

- Camps
- GMA Rivers
- GMA Roads
- Units
- National parks
- National border

Layer Properties:

Visible On Open:	☑	Label Bold	□
Custom Name:	Camps	Label Color:	black
Color:	dark green	Can Zoom/Blink:	☑
Symbol Style:	Triangle	Blink Field(s):	CampName NumStaff
True Type Font:	Wingdings	Renderer:	Class breaks
Character:		Rend Query:	qmap_NumStaffByCamp
Auto Find:	field patrol obs	Field With Values:	NumStaff
Symbol Size:	10	Num Groups:	5
RelateTable:	(query)	Grouping Method:	equal interval
Relate To Field:	AcCampID	Color Method:	preset color ramp
Relate From Field:	CampID	Preset Color Ramp:	rainbow 10
Can Add Labels:	☑	Start Color:	
Label Field(s):	CampName Nu	End Color:	
Label Add Init	□	Use Hatch Patterns:	□
Label Font:	Arial Narrow	Smallest Symbol Size:	15
Label Size:	8	Biggest Symbol Size:	25
		Custom ID Function:	□

Record: 8 of 16

Start a new record for your new form by clicking the *Data Entry* button or using the navigation buttons at the bottom. Or you can duplicate an existing map definition with the Duplicate button. Enter a name for your map definition.

In the first option box, select whether your map definition is for an Interactive Map/ Slide Template, or a custom map form. Most of the time you will be creating an interactive map. If your map definition will be used for a customized map form, then the name of your Map Definition should be the same as the name of the form.

If your map is going to be used to as part of a Slide Template, then set the appropriate options in the Slide Template settings box. Usually, you will want the maps created on your PowerPoint slide to be zoomed in to a particular area (as opposed to showing all of Zambia). If so, check the Pan to Marked Features box and enter the name of the layer which should be used for panning.

Note that until you start adding layers to your Map Definition, you won't see anything in the In Layer combo box. However after you add layers, you will see those layers which are joined to lookup tables.

For example, if your Slide Template is going to show a variety of summaries of field patrol results, and you only want it to show for a single area at a time, then you would want the map to pan to the selected Unit. In this case selected means the Unit(s) which have been ticked in the Filter Manager.

Next, click the *Add* button and start adding the layers you want to be available in you map. For an interactive map, not all of the covers have to visible initially. In fact usually you will only want 2-3 covers to be visible initially. However you can still add other covers in case the user wants to show more features in background. For a Map Definition which will be used on Slide Template, the user doesn’t have the option of selecting which layers to make visible, so only add those layers which you want to appear on the slide.

The covers will be loaded in the same order that they appear in the box on the left, but the stacking order can be changed by using the Up and Down buttons. The covers at the bottom of the list box will appear beneath all the other layers. Hence to prevent hiding you usually want the polygon layers at the bottom and the point and line coverages near the top. Note however you can also make polygon layers transparent, so there are times when you will want polygon layers at the top of the list (e.g., if you only want the polygon border to be visible but you want to make it thicker and on top of all the other layers)

After you have added all of the covers you want in your map, set the properties for each cover. For Interactive Maps, many of these properties can also be changed by the user after the map opens.

Visible on open – determines whether the layer will be visible when the map first opens. On Interactive Maps, the user can click the Layers button on the toolbar to turn the visibility of different layers on or off. However on map definitions which will be used on Slide Templates, only those layers which are set to be visible on open will appear on the slide.

Custom name – this is the name that will appear in the active layer combo box on the map toolbar, as well as the text on the legend created in PowerPoint. Usually the name of the cover is fine, however you may want to enter a custom name, particularly for layers that are classified. For example, “Trophy Locations” is a more user-friendly name than “5km Grids Saf Centers”.

Color – the initial color of the layer. On Interactive Maps this can be changed by the user by clicking the Layer Options button on the toolbar.

Symbol Style – additional options for the initial display of the layer. If it is a point layer, you can select which type of symbol you want used (e.g., circle, square, triangle, etc.) If it is a line layer, you can select if you want the line displayed with dashes or dots. If it is a polygon layer, you can select which hatch pattern you want used. On interactive maps, this setting can also be changed by the user.

True-Type Font – if the layer contains point features, and you have selected True-Type marker as the symbol style, then type the name of a True-Type font installed on your system. For example, “Wingdings”. See also Using Symbols from the Custom ADMADE True-Type Font below.

Character – if the layer contains point features, you have selected True-Type marker as the Symbol Style, and have typed in the name of a True-Type font, then select one of the characters from the font in this combo box.

Auto Find – if the layer contains point features, you have selected True-Type marker as the Symbol Style, you have entered “ADMADE” as the True-Type font, and you want Access to pick which character to use to display the features (instead of always using the same symbol) based on which records are Marked in one of the lookup tables, then select one of the lookup tables listed in this box. See Using Symbols from the Custom ADMADE True-Type Font below for more details.

Symbol size – determines the thickness of the outline (for a line or polygon cover) or the size of the symbol (for point layers), in points. The ‘normal’ setting for line and polygon layers is 1, but you can make it 3 or 4 if you want the features to have a thicker outline. The ‘normal’ size for point features is 5, but this can be made bigger. When used as an interactive map, this setting can also be changed by the user after the map is opened. Note that if this is a point layer which is being classified based on a query, then this setting will be ignored in favor of the Largest Symbol Size and Smallest Symbol Size properties (see below).

Relate table – this is the name of a table which will be joined to the coverage. Normally this will be one of the lookup tables or a query. If you want the layer joined to a query so that the features can be classified, select "(query)" from the combo box and type in the name of the query in the Rend Query property.

Relate to field – this is the name of a field in the shape file attribute table that will be used to join to the Access table/query.

Relate from field – this is the name of the field in the Access table or query which will be used to join to the shape file. If you're creating a join to a query, this should be the name of the first field in the query. See Designing a Query to Classify Map features below.

Can add labels – When used as an Interactive Map, this box determines whether the user will be able to add labels to the features in this layer by clicking on the Add Labels button on the toolbar. For some layers, such as Units and Camps, you probably want your users to be able to add labels. However it would be silly to try to add labels to features in layers such as roads and rivers, for which no names exist in either the shape file or Access. For grids, you might want to enable labeling so the user can add a label for the grid number or if classified another numeric value (e.g., representing number of poachers).

Label fields – the name of the field(s) which contain data which can be used for adding labels. If there is more than one field which contains valid data for labeling, separate each one with a space. If the cover is going to be joined to a query, then the label fields should be set to fields #2 and #3 (which in the case of a query designed for classifying with class breaks will contain (2) a string identifying the feature and (3) a numeric value for the feature)

Note: if the map definition is used as part of a Slide Template, and you have Label Add Init checked, then the *first* field listed in the Label Fields box will be used to create the initial labels.

Label Add Init – check this box if you want the layer to be labeled when the map initially is opened. Note that on maps for Slide Templates, this box must be checked if you want the features for this layer to be to be labeled in the map in PowerPoint.

Label Font – the default font for labels. If left blank then Arial will be used.

Label Size – the default size for labels in points. 8 or 9 is usually a safe size.

Label Bold – whether the labels should be initially bold

Label Color – the color of the label

Can zoom/blink – check this box if you want the user to be able to use the Find and Blink buttons on the toolbar with this layer. Applies only to Interactive Maps

Blink fields – the name of the field(s) that the user should be able to use to select features with the zoom/blink buttons. If there is more than one field, then separate them with a space.

Renderer – if the features in this layer will be classified (i.e., colored), then select the type of renderer in this box. The options are class breaks, pie chart, bar chart, or none. Note: pie and bar charts do not work reliably with Access 97.

Rend Query – if the features in the layer will be classified according to the results of a query, then enter the name of the query in this box. See Designing a Query to Classify Map Features for more details on the required specifications for the query.

Field With Values – if the layer is going to be classified according to class breaks, then enter the name of the field in the table/query which contains the numeric data which will be used to classify the features. If the layer will be classified using a query, this will normally be the third and final field in the query.

Num groups – for class breaks classifications, enter the number of initial numeric groups to create. Note the user can change this color later using the classification options button on the map tool bar. This setting has no effect for pie and bar charts, for these classifications MapObjects will create pars/pie slices based on the number of fields in the query/table.

Grouping method – for class breaks classifications, enter the method which should be initially used to create the color groups, equal interval (divisions based on the minimum and maximum values of the field) or quantile (divisions calculated based on the number of features in each group). Note the user can change this setting later using the classification options button on the map tool bar.

Color method – select the method for coloring the features. You can either use a Preset color ramp or linear colors

Preset Color Ramp – select one of the available preset color ramps.

Start color – for linear coloring, enter the start color. Note the user can change this color later using the classification options button on the map tool bar. This setting has no effect for pie and bar charts

End color – for class breaks classifications, select an end color. Note the user can change this color later using the classification options button on the map tool bar. This setting has no effect for pie and bar charts

Smallest symbol size – if this is a point layer which is being classified, then enter the size in points of the smallest symbol. If you don't want the size of the symbol to represent the value, then enter the same number for the smallest and largest symbol size.

Largest symbol size – if this is a point layer which is being classified, then enter the size in points of the largest symbol.

Use hatch patterns – determines whether you want to use hatch patterns in the classification (e.g., vertical stripes, diagonal lines, etc.) Hatch patterns are most useful if you know that the map will eventually be printed on a black and white printer. If you use hatch patterns, you probably want to also use a linear color ramp, where the first and last color are both black, in order to force all groups to be in black.

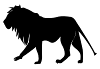
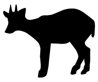
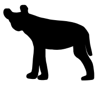
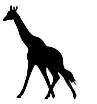
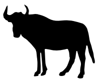
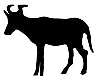
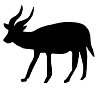
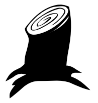
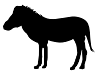
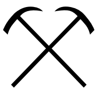
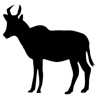
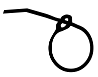
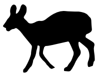
Custom ID function – some interactive maps have a custom function when the ID tool is used. This requires some slightly advanced programming knowledge. If your map is one of these, click this box.

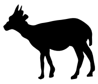
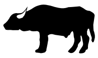
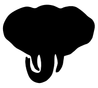
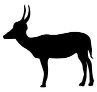
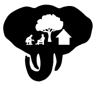
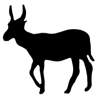
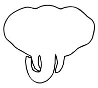
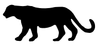
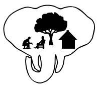
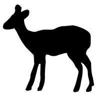
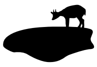
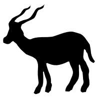
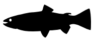
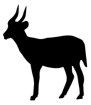
Your map definition is now finished. If you want this Map definition is going to be the basis of an Interactive Map, all that remains is to add it to the Menu System using the Menu Manager. If you want the map definition to be part of a Slide Template, then go to the Slide Templates form and create a new element for your map. Remember, if your map definition is classified using a query which is filtered based on the Marked field in the lookup tables, be sure to enter the correct filter fields when adding your map definition to either the menu system (to make an interactive map) or Slide Template.

Using Symbols from the Custom ADMADE True-Type Font

You can use symbols from the custom ADMADE font on your maps. Of course you can only uses these symbols to classify point covers, such as camps or the center points of grids. The ADMADE True-Type font has symbols for each of the common safari animals, (e.g.,) as well as other common observations from field patrols and safari hunts, such as timber cutting, water holes. The table below shows the complete list of characters in the ADMADE font.

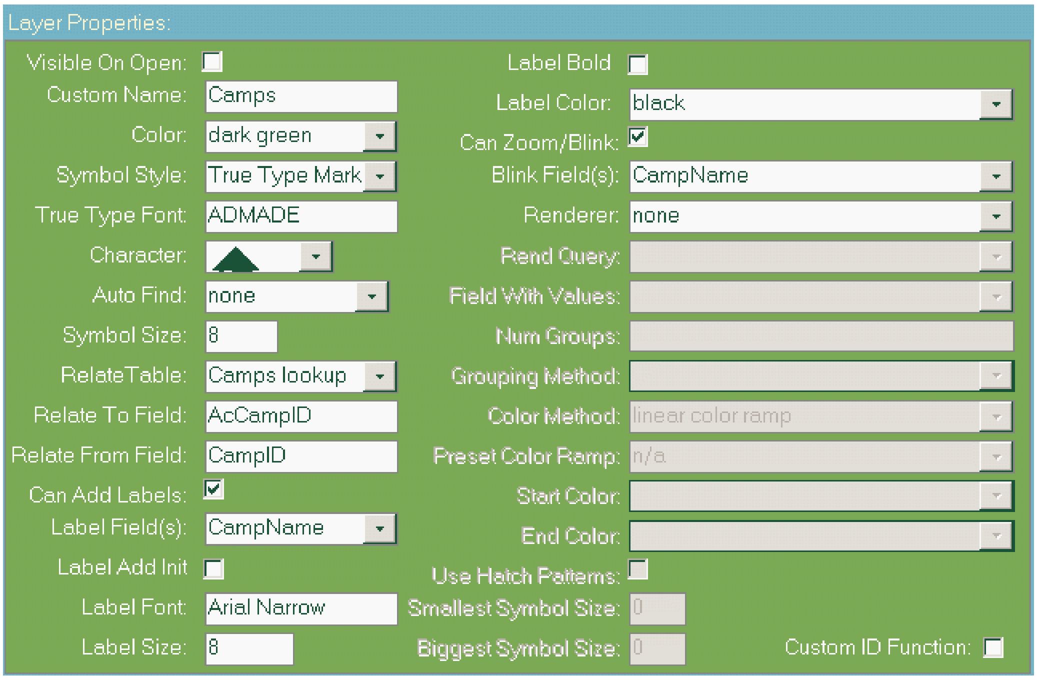
Note: you will only see the animal symbols in the table below if you have the ADMADE font installed on your system. To install it, copy the ADMADE.TTF file to your \WINDOWS\FONTS directory.

Symbol	Index	Key	Character	Symbol	Index	Key	Character
Lion	33	!		Klipspringer	53	5	
Rhino	34	"		Hyaena	54	6	
Giraffe	35	#		Hippo	55	7	
Wildebeest	36	$		Hartebeest	56	8	
Warthog	37	%		Kudu	57	9	
Sitatunga	38	&		Poacher	58	:	
Sable	39	'		Fire	59	;	
Roan	40	(		Stump	60	<	
Zebra	41	)		Mining	61	=	
Tsessebe	42	*		Snare	62	>	
Grysbok	43	+		Hut/Camp	63	?	
Eland	44	,		Hunter	64	@	

Duiker	45	-		Circle	65	A	
Buffalo	46	.		Logo1	66	B	
Impala	47	/		Logo 2	67	C	
Reedbuck	48	0		Logo 3	68	D	
Leopard	49	1		Logo 4	69	E	
Oribi	50	2		Waterhole	70	F	
Lechwe	51	3		Fish	71	G	
Puku	52	4					

Because the ADMADE font is a standard Windows True-Type font, it can be used by any Windows program. For example you can use the symbols in Word processing documents. You can also use the font in other graphing or desktop mapping programs that can make symbols from True Type fonts. For example, ArcView can use symbols from a True-Type font for a graduated point classification of themes.

To use one of the characters from the ADMADE font to display point features on your map, you specify it in your Map Definition. For the point feature layer properties, select *True-Type Font Marker* as the symbol style. Then type the name of one of the font in the True Type Font box (e.g., “ADMADE”) Then select one of the characters from the combo box. If the layer is being classified, then the size and/or color of the symbols will vary based on the color method and smallest/largest symbol size properties



Layer Properties:

Visible On Open:	□	Label Bold	□		
Custom Name:	Camps	Label Color:	black		
Color:	dark green	Can Zoom/Blink:	☑		
Symbol Style:	True Type Mark	Blink Field(s):	CampName		
True Type Font:	ADMADE	Renderer:	none		
Character:	▲	Rend Query:			
Auto Find:	none	Field With Values:			
Symbol Size:	8	Num Groups:			
RelateTable:	Camps lookup	Grouping Method:			
Relate To Field:	AcCampID	Color Method:	linear color ramp		
Relate From Field:	CampID	Preset Color Ramp:	n/a		
Can Add Labels:	☑	Start Color:			
Label Field(s):	CampName	End Color:			
Label Add Init	□	Use Hatch Patterns:	□		
Label Font:	Arial Narrow	Smallest Symbol Size:	0		
Label Size:	8	Biggest Symbol Size:	0	Custom ID Function:	□

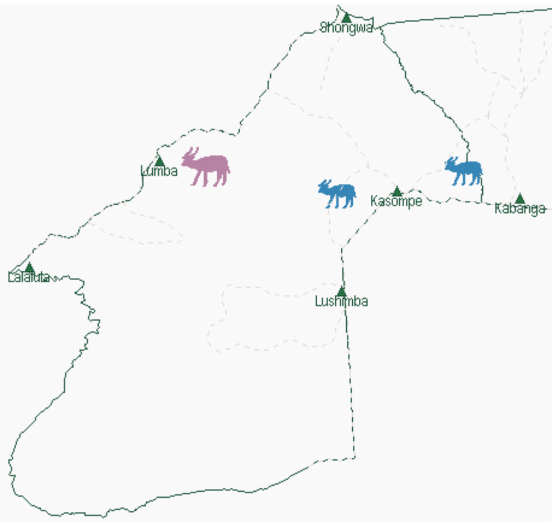
Auto Find

You can use the Auto Find feature to have ADM pick which character from the ADMADE font to use when drawing the map features. Auto Find will pick a character based on the status of one of the lookup tables. There are three lookup tables that have been set up to work with the Auto Find feature: species, safari hunt observations, and field patrol observations. Basically what happens is that when it comes time to display the point features on your map, ADM will go to the look up table and see which record or records are Marked (meaning the Marked field = True). If Access finds that one and only one record in the lookup table has Marked = true, then it will use the character for that record (in the AutoCharIndex field) when drawing the points on the map. If it finds that no records are marked or more than one record is marked, it picks a default character (a circle).

The Auto Find feature is quite handy when your map definition is going to be used to display occurrences of an individual species or individual observation. For example, you might want a map that will use animal symbols from the ADMADE font to show the location of animal sightings from field patrols. To create this kind of map, you would first create a query to sum up the number of animals sighting per grid, and use the Marked field from the Species lookup table to allow the user to select which species will be output in the query.

Then you would start or modify a map definition and add the point version of the national grid cover, which only has the center points of each grid. Then, in your map definition, you would select “(query)” as the relate table and enter the name of your query as the Rend Query property. Finally you would select “True-Type font” as the symbol style, “ADMADE” as the True-type font, and “Species” as the Auto Find lookup table.

Thus, when the user opens the map, assuming that they mark one and only one species in the Filter Manager, then the map will use the appropriate symbol for that species. And since the query is also filtered based on the lookup table, the number of sightings per grid will be for only that species. If the user selects the Filters button and marks a different species in the Filter Manager, then the map will be updated including the animal symbol. If they mark more than one species, they will see circles instead of an animal symbol.



With Auto Find, ADM will pick the appropriate symbol used to display the symbols based on which species is Marked in the lookup table. Here we see a map of the locations where Sitatunga were hunted

Using the Auto Find feature is also handy when you want to have a repeating map on a Slide Template using a point coverage. For example, you may want a map (showing location of animal kills) in your Slide Template to repeat once for each species marked in the Filter Manager. When ADM gets ready to create the maps, it will first ask the user to select which species maps should be created for. Then it will set Marked = false for all records, then it will turn them back on one-by-one and create the individual maps. Hence each time a map is created and send to PowerPoint, AutoFind will examine the lookup table and pick the appropriate symbol based on whichever record is currently marked.

Like all point features, you can also use color ramps and/or sizes to show values behind the symbol. Simply set the Smallest Symbol Size and Largest Symbol Size properties for the map layer. The sample above uses both colors and sizes to show how many animals were hunted at the grid locations.

Designing a Query to Classify Map Features

If one or more of the layers in your map definition is going to be classified using values from a query, your first step is to prepare the query. The query you create must follow *exact* specifications in order for it to work with the ADM Map Objects module. These specifications are explained as follows:

- The very first field of your query should be a *unique* numeric field which contains values which will be used to join to the shape file. For example the first field could be the UnitID, HblockID, Universal Grid Number, etc., depending on which features the query will classify. The name of the field (i.e., column heading) does not matter.

  Note: shapefile coverages for the 5km grids use a universal 6 digit number to uniquely identify each grid. The first two numbers are the ID number for the Unit or Hunting Block, and the last four numbers are the actual grid number. For example, grid number 140042 would be grid 42 in Mumbwa Unit (whose UnitID number is 14). Thus if you're creating a query which will be used to classify features from the field patrol or safari hunting grid coverages, then the first field in your query should contain a formula which returns this unique 6 digit number.

- The second field of your query must contain a string or number which can be used for adding labels. For example, if your query is calculating the total hunting revenue for each Unit, and is going to classify the Units coverage, then you probably want the second field of the query to be the Unit name. The user will be able to use this field to label the coverage.

  If you don't want the user to be able to label the features in the coverage, then leave the second field in the query blank (i.e. use a calculated expression, such as *BlankName: ""*). Also you can uncheck the Can Add Labels box in the map definition.

- If your query is going to classify features using class breaks, then the third and final field of your query should be a numeric field that represents the numeric variable of interest (i.e., number of poachers caught, income, average search time, etc.). The name of the field (i.e. column heading) should be the text which you want to appear in the combo box on the map toolbar and in the legend. You can change the name of a field in a query by entering the name followed by a colon followed by the field or expression in the top row of the query design grid (ex: Unit: UnitID).

- If your query is going to be used to classify the features in the coverage using a pie or bar chart, then the 3rd, 4th, 5th, etc. field(s) should contain the numeric values that will be used to create the pie/bar charts. You can use up to 10 numeric fields in your query. These could represent years, types of employees, or whatever. Often you will use a crosstab query to produce these fields. Note that the field names will appear in the legend of the map. *Note: bar and pie charts do not work well with this version of Map Objects.*

Other than these requirements, there are no restrictions on the design of queries you use to classify map features. You can and probably will want to use lookup tables (so you can filter it based on the boolean Marked field of the lookup table), use left or right sided joins, crosstab queries, nest queries, etc., as long as the final product returns the fields listed above in the specified order. Give your query with a name which ties it to your Interactive Map or slide template.

Adding a New GIS Cover

Many of the standard national GIS covers need to make maps in ADM have already been prepared. These include:

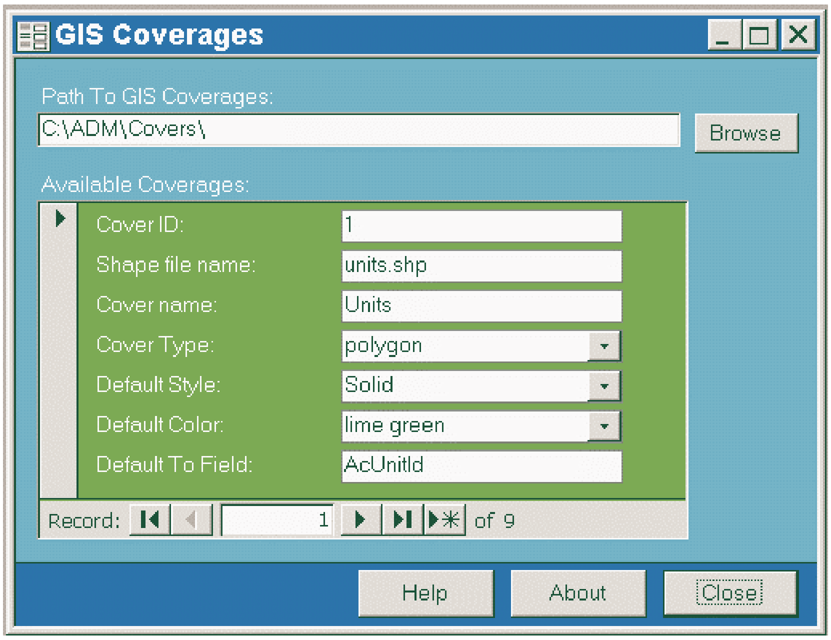
- National border
- Units
- Hunting blocks
- GMAs
- Scout Camps
- National Parks
- GMA Roads
- GMA Rivers
- Field patrol grids
- Safari hunting grids
- Major rivers
- Major roads
- Districts
- Provinces
- Provincial capitals
- Railroads
- Utility lines
- Airports
- Villages
- Wetlands

However inevitably there will be additional coverages that will be wanted as new needs arise. In order for covers to overlay properly and function the same way, all GIS covers used by ADM should:

- be saved in ESRI shape file format
- be in geographic coordinates (i.e., Lat-Long)
- contain features for the whole of Zambia
- be saved in the directory listed in Program Options
- if it's going to be joined to a table in Access, have a field with values that match on of the primary keys in a lookup table

See also Tech Notes on Preparing Coverages for ADM.

After you have prepared a coverage according to the guidelines above, the next step is to enter the properties of your coverage in the GIS Covers form. From the Main Menu, select Admin => Mapping Setup => GIS Covers.



GIS Coverages

Path To GIS Coverages: C:\ADM\Covers\

Available Coverages:

Cover ID:	1
Shape file name:	units.shp
Cover name:	Units
Cover Type:	polygon
Default Style:	Solid
Default Color:	lime green
Default To Field:	AcUnitId

Record: 1 of 9

The property settings for each cover are fairly self-explanatory. A few notes:

- each coverage should have a unique ID number between 1 and 255
- the shape file name should include the shp extension, but not the path. ADM assumes that all shape files are located in the Path to GIS Covers box
- The Cover name will be the default name which appear in the active layer combo box on the toolbar on the Map Form. However can also be overridden with the Custom Name property of the map definition.
- Default To Field is the name of the field (item) in the shape file attribute table that will normally be used to join to an Access lookup table
- all of the default settings can be over-ridden when you enter properties in your map definition

Tech Notes on Preparing Coverages for ADM

ADM uses Map Objects© to create Interactive Maps and Slide Templates. Map Objects is designed to work best with GIS data in ESRI Shapefile format, so ArcInfo coverages need to be converted to Shapefiles. Also, because Interactive Maps present data from Access tables which store information for all areas of the country, all GIS layers have been converted into national coverages and given additional ID fields that let them work with the tables in Access.

The following are a few additional tech notes about converting ArcInfo coverages to work with ADM.

- National covers should be projected in geographic coordinates (e.g., latitude-longitude) in order to overlay properly with the other layers. There is no reason why another large-scale projection system could not be used for the covers, such as Albers, however geographic coordinates were chosen because a lot of other spatial data, including data from other agencies, is available in lat-long. Also you can re-project Lat-Long covers on the fly in ArcView. UTM, which is the projection of most of the 1:250,000 maps available from the geographic department in Lusaka, is not appropriate for a national coverage because UTM covers can only span 6 degrees of longitude, which is not big enough to capture all of the ADMADE areas.

- There are two coverages for the 5 km grids, one for field patrol data (and other data which is based on Units), and another for safari hunts (and other data which is based on hunting blocks). This was necessary because some Units have more than one hunting block (e.g., Mumbwa, Chikwa) or more than one base map (e.g., Munyamadzi). If there weren't two versions of the grid coverage, then any maps created for these areas would be faulty. If you want to make a map which shows both safari hunting and field patrol data, you need to either limit yourself where the maps are the same, or make a transformation table to convert grid numbers from one base map to the other.

- In addition, the national 5 km grid coverage has a polygon version and a point version. This was necessary because ArcView shapefiles, unlike ArcInfo covereages, can only contain one feature class. The point version of the 5 km grid coverage contains the label point (i.e., center) of each grid, and is used for interactive maps which present grid data using point symbols (such as the single-species maps or field patrol observations). Thus there are a total of 4 national coverages for grids, 2 for field patrols (a polygon and a point coverage) and 2 for safari hunts.

- Most of the coverages digitized Nyamaluma are separated for different Units, and saved in UTM coordinates. The general steps to transforming these coverages for use with ADM is as follows:

    1. Create a backup copy of the coverage. **Make certain there are no label errors or slivers** (i.e., each polygon has one and only one label point) to avoid problems later.

    2. Delete any extraneous items from the attribute table, such as Address and Address_Id (remember when appending coverages in ArcInfo, the items in each cover must be identical, same names, type, and order)

    3. For polygon coverages, add two numeric items called AREAM2 and PERMIMM, and copy the values of the Area and Perim items into these new items. This is needed because when the cover gets projected into geographic coordinates, the values of the Area and Permin items will be converted to degree coordinates which are meaningless. So by copying their contents into new items before using the transformation command is issued, we preserve the area and perimeter values of each feature and enable analyses based on feature area and perimeter. Note: there is an SML macro (@addarea) which adds these items and fills them automatically.

    4. If converting a grid coverage, add an additional numeric item called GRID_NUM and copy the values of the User ID into this new item. This is needed because when appending all of the covers together into one big national coverage and then cleaning it, the User Ids will be altered.

5. Transform the cover into geographic coordinates. You can use the @convll SML macro to help do this.

6. Add another item which will be used to join the attribute table to the lookup table in Access. For example, you could add an item called ACVAGID, which corresponds to the ID value of the lookup table for VAGs. Other standard item names are ACUNITID (the ID number of a Unit in Access) and ACHBLOCKID (the ID number of a Hunting Block). Enter the correct values into these items using ArcEdit or Tables.

7. Once all of the individual covers have been prepared as stated above, combine them together using the ArcInfo *Append* command. Note that the Append command requires that all coverages have *identical* items (same names, same data types, same order). Also note you can enter the covers you'd like to append using a text file rather than typing each one in each time you use the command.

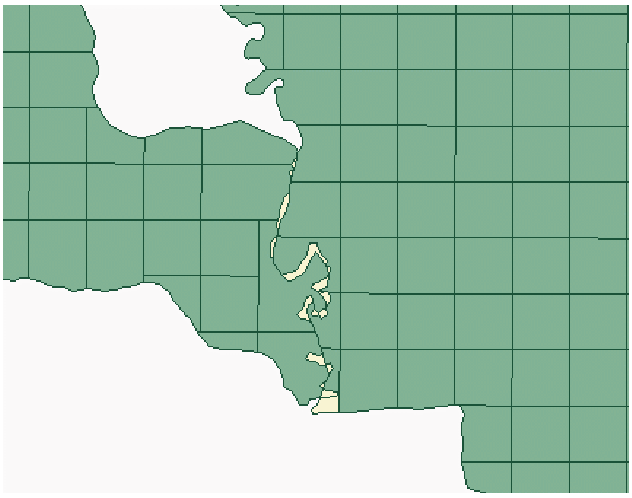
8. If you're creating a polygon coverage, clean the new combined coverage, using a small fuzzy tolerance (such as 0.00001), as follows:

```
(C:\)[ARC]CLEAN UNIT_ALL UNIT_ALL1 0 0.0001
```

Note: failure to use a small fuzzy tolerance as shown above will likely result in collapsing of polygons during the clean process which may, depending on the location of nearby label points, will result in grids getting the wrong grid number value.

9. If converting a Grid coverage, then you also need to create a point version of the ArcInfo coverage. To do this, make a copy of the *original* appended coverage (not the one which has been cleaned) and use the BUILD command on the copy with the POINT option.

10. Open the new coverage in ArcView. Select any slivers created by the append command. This can be done most easily by opening the theme attribute table and selecting those records where one or more items is set to 0 or NIL. After verifying in View that only slivers are selected, go back to the theme attribute table, toggle the selection, then go back to the view and use the Convert to Shapefile command to save only those selected features in a new shape file.



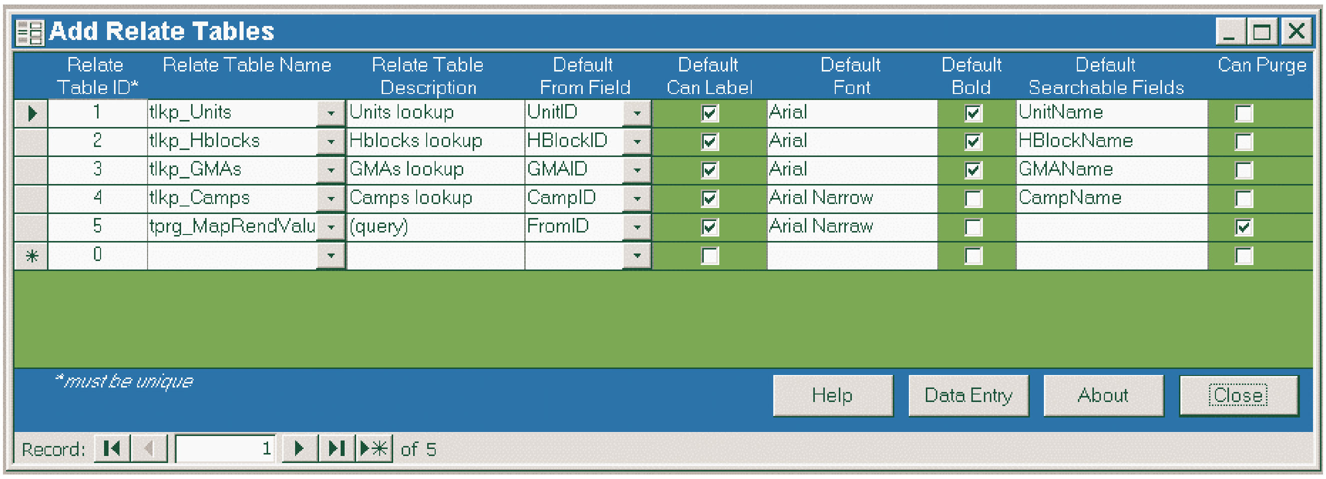
11. If converting a grid coverage, create a new item called ACGRIDNUM for the Universal Grid Number. This item should be a 6-digit numeric item, whereby the first two number are the ID of the Unit or Hunting Block (depending on which 5km grid covers you appended), and the final 4 digits record the number of the grid. Use the calculate command on the field menu to enter values in this field. This must be done for both the point version and the polygon version of the grid coverage.

12. To reduce disk space, you can delete any un-needed fields from the shapefile attribute table, such as Area, Perimeter – which are both meaningless because they're in degree coordinates, and the ArcInfo ID fields.

13. Put the new/updated shapefile in the C:\ADM\COVERS directory, which all the other shape files used by ADM.

Adding a New Add Relate Table

Before you can join a GIS cover to a table in Access, you must enter the properties of the Access table. From the Main Menu, select Admin => Mapping Setup => Add Relate Tables.



Add Relate Tables

	Relate Table ID*	Relate Table Name	Relate Table Description	Default From Field	Default Can Label	Default Font	Default Bold	Default Searchable Fields	Can Purge
▶	1	tlkp_Units	Units lookup	UnitID	☑	Arial	☑	UnitName	□
	2	tlkp_Hblocks	Hblocks lookup	HBlockID	☑	Arial	☑	HBlockName	□
	3	tlkp_GMAs	GMAs lookup	GMAID	☑	Arial	☑	GMAName	□
	4	tlkp_Camps	Camps lookup	CampID	☑	Arial Narrow	□	CampName	□
	5	tprg_MapRendValu	(query)	FromID	☑	Arial Narrow	□		☑
✳	0				□		□		□

*must be unique

Record: 1 of 5

The fields for each add-relate table are pretty self-explanatory, but a few notes:

- Each relate table must have its own unique ID number between 1 and 255
- Relate Table Description – this is the text which will appear in the combo box on the Map Definition Properties dialog
- Default From Field is the name of a field in the table which normally will be used to join to shape files. This is usually the primary key field
- Default Can Label specifies whether you want the user to be able to label the features in the coverage that will be related to this table.
- tprg_MapRendValues is a special add-relate table that is only used as an intermediary when a shape file is joined to a query. Don't delete it or change its settings.
- 'Can Purge' determines whether the VBA code can delete the records from the add-relate table. You would only want to enable this if the table contains dynamic values that are updated or appended based from a query (such as tprg_MapRendValues). In general, you do not want to enable Can Purge.

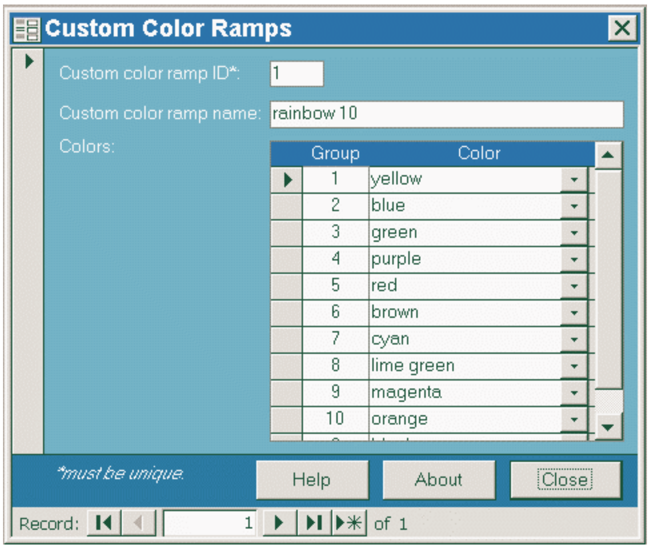
Creating a New Preset Color Ramp

Preset color ramps are used in Map Definitions where features in one of the layers are classified according to numeric values. Using a preset color ramp has certain advantages over linear color ramps, including:

- The colors in each group can be made distinctive and easy to distinguish. This is particularly important in maps that will be printed on a plotter for group presentations.

- The meaning of different colors can become standardized. Once users will learn that yellow always means a low value, and red is always a high value, comprehending the map will be made easier and faster.

Creating a new preset color ramp is easy. Select Preset Color Ramps from Admin – Mapping Setup on the Main Menu. The interface for the data entry form is self-explanatory:



Custom Color Ramps

Custom color ramp ID*: 1

Custom color ramp name: rainbow 10

Colors:

Group	Color
1	yellow
2	blue
3	green
4	purple
5	red
6	brown
7	cyan
8	lime green
9	magenta
10	orange

*must be unique.

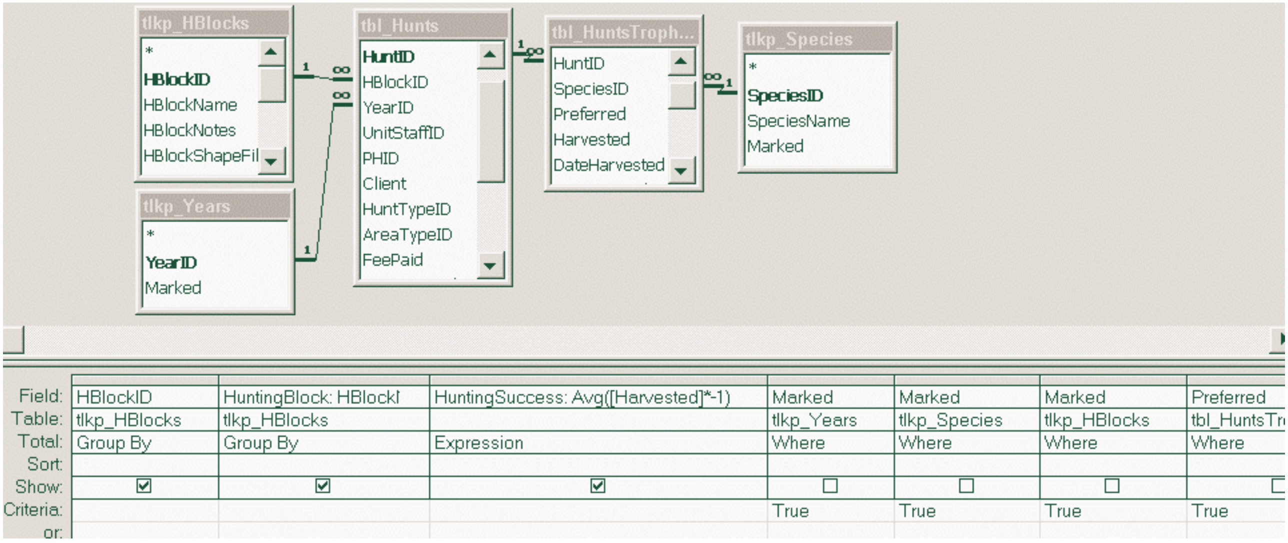
Record: 1 of 1

Exercise: Creating an Interactive Map

In this exercise, we will create a new Interactive Map which will classify (e.g., color) hunting blocks according to hunting success. Hunting success is calculated by taking the number of animals which were successfully shot divided by the number of animals which were preferred. This information will come from the tbl_HuntsTrophies table. We want our map to function in the following ways:

- the user should be able to select which species that will be used for the calculation (so for example the map can just show the hunting success for just a single species or just the big five, or all species combined
- the user should be able to select which hunting blocks are included
- the user should be able to select which years are used in the calculations
- only those records where the SAFLICE data form was filled out correctly should be used
- the user should be able to add labels to see the exact percentage of hunting success for each hunting block
- the user should be able to have the option of displaying roads, rivers, and camps as part of the background

We know that we need to use a class breaks classification (as opposed to a pie chart or bar chart classification), and that the cover that will be classified will be hunting blocks. Furthermore, we will need to join the hunting blocks attribute table to a query, and that query should have the lookup tables for species, hunting blocks, and years in it. We also know that a query which is used for a class breaks classification should have three fields total. The first field will be the field used to join to the shape file (in this case the Hunting Block ID), the second field will be a text field used for labeling (in this case the hunting block name), and the third field will be a numeric value which will be used for classification and/or labeling. Hence our query in design view should look like this:



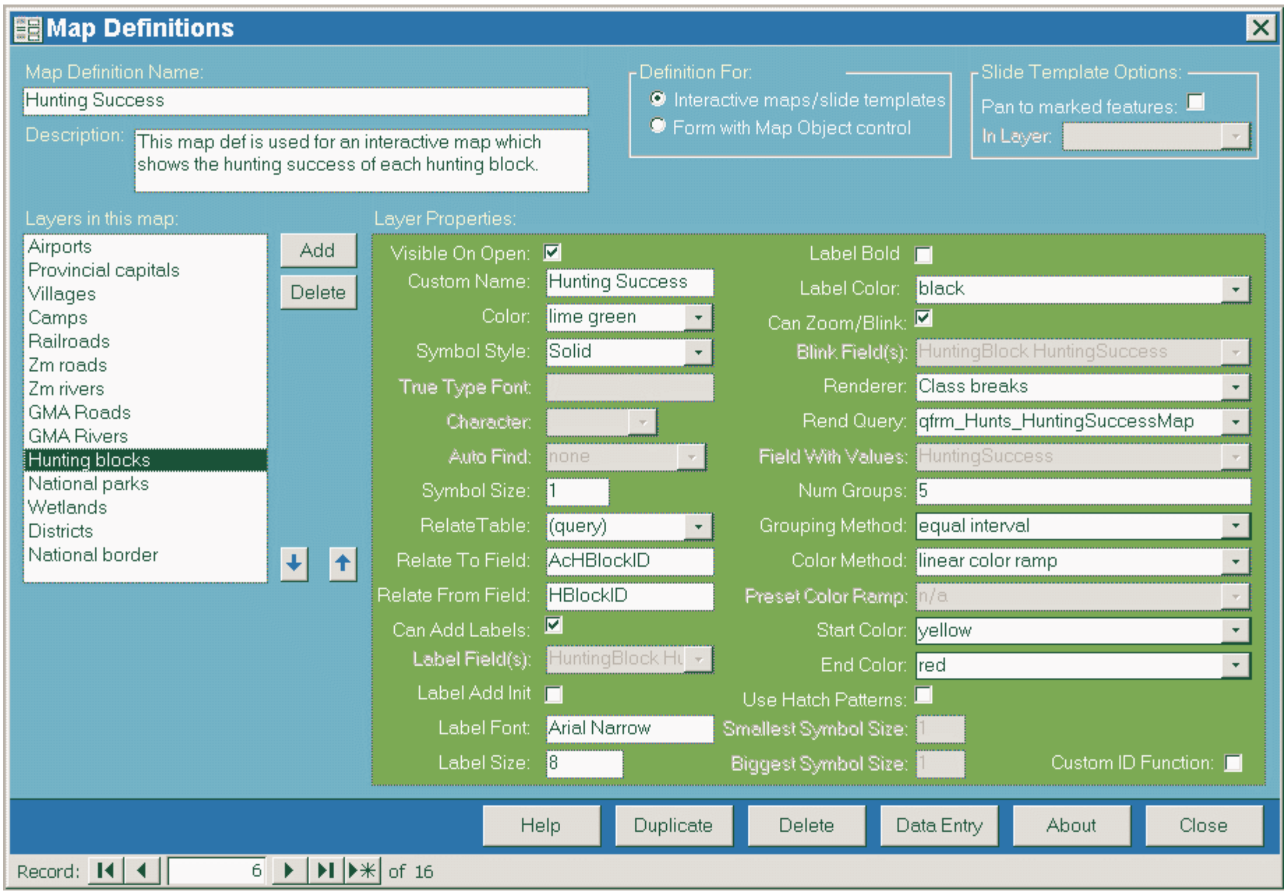
Field:	HBlockID	HuntingBlock: HBlockI	HuntingSuccess: Avg([Harvested]*-1)	Marked	Marked	Marked	Preferred
Table:	tlkp_HBlocks	tlkp_HBlocks		tlkp_Years	tlkp_Species	tlkp_HBlocks	tbl_HuntsTr
Total:	Group By	Group By	Expression	Where	Where	Where	Where
Sort:							
Show:	☑	☑	☑	☐	☐	☐	☐
Criteria:				True	True	True	True
or:							

Note the following features about this query:

- Although there are quite a few fields in the design grid, only three fields have a check in the *Show* row, so only three fields will appear in the output.
- the first column is HBlockID, this will be used to join with the shape file which has an identical field called AcHBlockID.
- We have used alternate column headings (i.e., field names) for the second and third field in the query. Instead of the second field being called 'HblockName', we have called it "HuntingBlock". This more user-friendly name will appear in the combo box in the map toolbar. Likewise the heading for the third column is "HuntingSuccess", instead of something like "Expression01". Note however our column heading don't have spaces, which is general good practice and in this case mandatory because MapObjects has problems with field names which have spaces.
- Because we want to use an aggregate domain function in this query (i.e., average) we have displayed the totals row in the design grid.
- The expression we are using to calculate hunting success is Avg([Harvested]*-1). Harvested is a boolean field, with values of 0 or –1. Some of those records will be 0's, and some will be -1's. By taking the average of the negative value of this field, we have in effect calculated the average hunting success.
- The lookup tables for Years, Species and Hunting Blocks are part of the query, and the Marked field for these tables are included in the design grid with a criteria set to True. This has the effect of only including those records from tbl_HuntsTrophies where the matching records in all three lookup tables have the Marked field = True. This will also allow us to employ the Filter Manager to easy change which records in the lookup tables have Marked = True
- There is also a criteria set for the Preferred field. If an animal was shot but not preferred, then for the purposes of this analysis we don't want to include it. Although this doesn't happen very often, it is possible to have a record for an animal that was not preferred but was nevertheless shot (as a substitute for another animal that wasn't found).
- Although we can't see it on the illustration above, another field in the design grid is "SAFLICEFilledInCorrectly". This field has a criteria expression of True, because we only want to include those records where the form was filled in correctly (meaning that the scout wrote down the species desired by the client *before* the hunt began).

Now that we have finished the query, we save it as qfrm_Hunts_HuntingSuccessMap. Note the naming convention used, we can tell immediately that this query is used for an interactive map.

Next, we create a map definition for our map in the Map Definitions form, which you can open from the Main Menu by selecting Admin => Mapping Setup => Map Definitions. We start a new record by clicking Data Entry, and then start loading the layers we'd like to be available for our map.



Map Definitions

Map Definition Name: Hunting Success

Description: This map def is used for an interactive map which shows the hunting success of each hunting block.

Definition For:
● Interactive maps/slide templates
○ Form with Map Object control

Slide Template Options:
Pan to marked features: □
In Layer:

Layers in this map:
- Airports
- Provincial capitals
- Villages
- Camps
- Railroads
- Zm roads
- Zm rivers
- GMA Roads
- GMA Rivers
- Hunting blocks
- National parks
- Wetlands
- Districts
- National border

Layer Properties:

Visible On Open: ☑
Custom Name: Hunting Success
Color: lime green
Symbol Style: Solid
True Type Font:
Character:
Auto Find: none
Symbol Size: 1
RelateTable: (query)
Relate To Field: AcHBlockID
Relate From Field: HBlockID
Can Add Labels: ☑
Label Field(s): HuntingBlock Hu
Label Add Init: □
Label Font: Arial Narrow
Label Size: 8

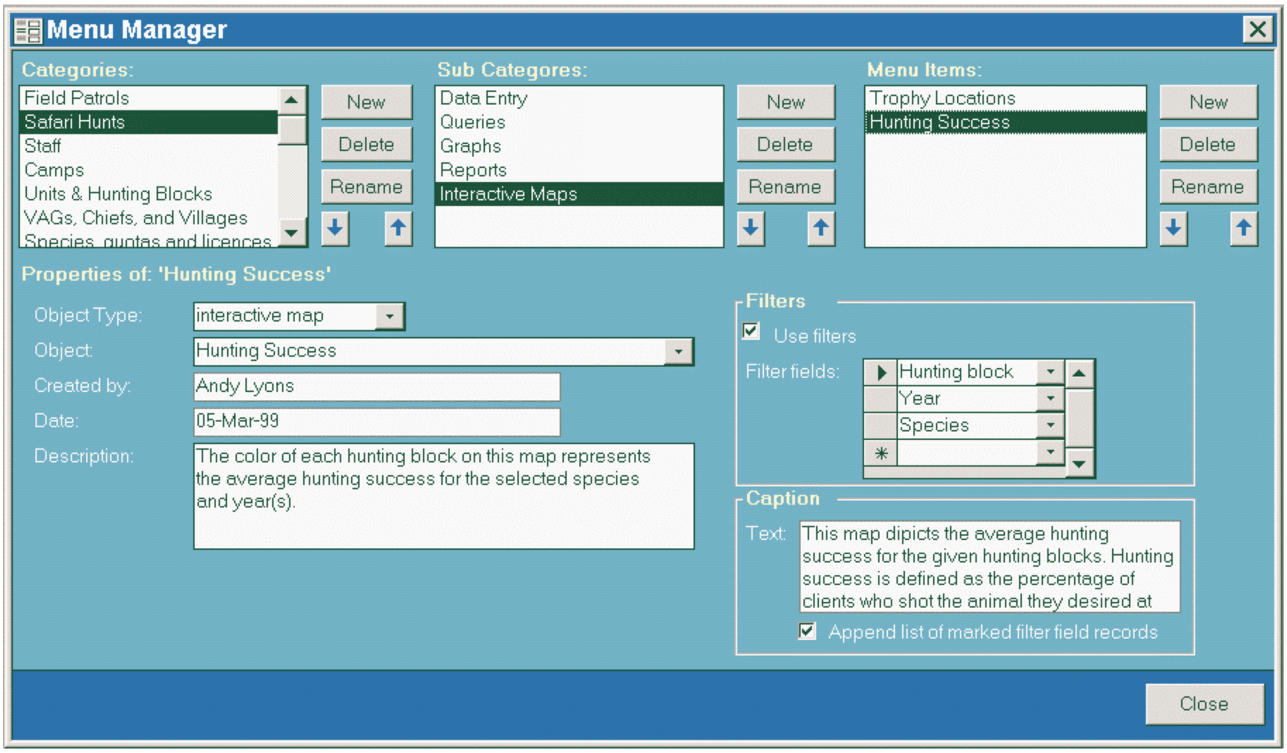
Label Bold: □
Label Color: black
Can Zoom/Blink: ☑
Blink Field(s): HuntingBlock HuntingSuccess
Renderer: Class breaks
Rend Query: qfrm_Hunts_HuntingSuccessMap
Field With Values: HuntingSuccess
Num Groups: 5
Grouping Method: equal interval
Color Method: linear color ramp
Preset Color Ramp: n/a
Start Color: yellow
End Color: red
Use Hatch Patterns: □
Smallest Symbol Size: 1
Biggest Symbol Size: 1
Custom ID Function: □

Record: 6 of 16

Note the following features of the properties for our map definition:

- We have added several layers, but we used the up and down arrows to stack them so that the point coverage (i.e., camp) and line covers (e.g., road and rivers) are on top of the polygon covers (so they won't be blocked).
- Although 14 covers are available for display on this map, only two of them will be visible when the map first opens. Namely the national border and the hunting blocks. The others can be made visible if the user clicks the *Other Layers* button on the map toolbar.
- For the hunting blocks layer, we have told it to create a join to a query, and we have entered the name of the query we created in the step above. We have also told it to use a class breaks classification (i.e., renderer) on this layer. The map code will automatically know to use the third field of the query as the classification field.
- We have checked the Can Add Labels box, so that the use can label the hunting blocks according to either the Hunting Block name (field 2 of the query) or the hunting success of that hunting block (field 3). We have also checked the Can Zoom/Blink box, so the user can select a hunting block name from the map toolbar and then flash or pan to it.
- We are using a linear color ramp, so the colors will go from Yellow to Red. However could have also used a Preset Color ramp if we want more control over the colors Access uses to draw the features.

Next we need to add our form to the Menu System. This will make it available on the Main Menu, and will also enable the Filter Manager and Caption features. Items are added to the menu system using the Menu Manager. The properties for this menu choice are as follows:



Categories:

- Field Patrols
- Safari Hunts
- Staff
- Camps
- Units & Hunting Blocks
- VAGs, Chiefs, and Villages
- Species, quotas and licences

Sub Categories:

- Data Entry
- Queries
- Graphs
- Reports
- Interactive Maps

Menu Items:

- Trophy Locations
- Hunting Success

Properties of: 'Hunting Success'

Object Type: interactive map

Object: Hunting Success

Created by: Andy Lyons

Date: 05-Mar-99

Description: The color of each hunting block on this map represents the average hunting success for the selected species and year(s).

Filters

☑ Use filters

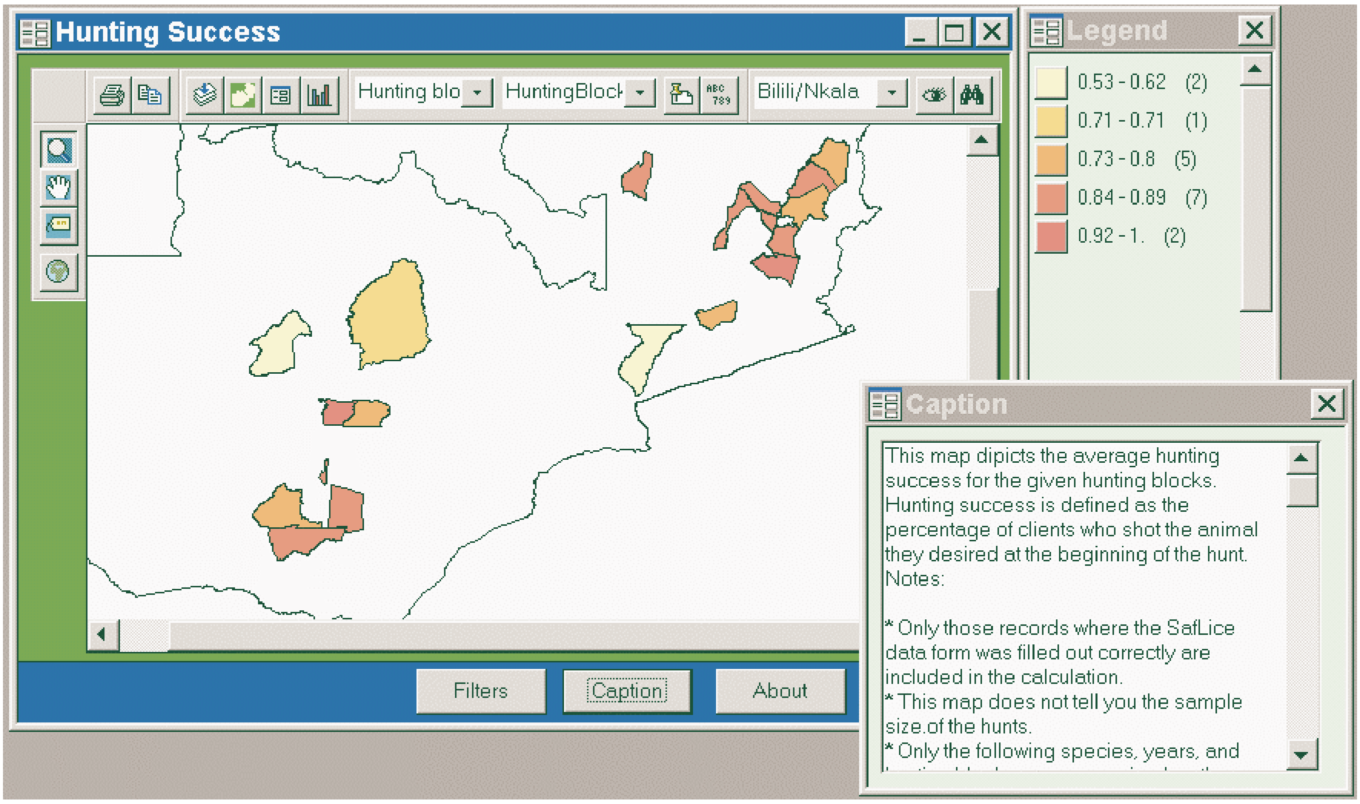
Filter fields: Hunting block, Year, Species

Caption

Text: This map dipicts the average hunting success for the given hunting blocks. Hunting success is defined as the percentage of clients who shot the animal they desired at

☑ Append list of marked filter field records

Once we close the Menu Manager, our new map form should be on the Main Menu and ready for use. The Caption, Filter buttons, and all the Map toolbar buttons should work fairly well.



Legend

- 0.53 - 0.62 (2)
- 0.71 - 0.71 (1)
- 0.73 - 0.8 (5)
- 0.84 - 0.89 (7)
- 0.92 - 1. (2)

Caption

This map dipicts the average hunting success for the given hunting blocks. Hunting success is defined as the percentage of clients who shot the animal they desired at the beginning of the hunt.
Notes:

* Only those records where the SafLice data form was filled out correctly are included in the calculation.
* This map does not tell you the sample size of the hunts.
* Only the following species, years, and

Chapter 4: Maintaining ADM

Like any MS Access database, the ADM needs to be periodically maintained for best performance and data security. You may also need to install ADM on a new machine, or update the files on a different machine with the most recent data. This chapter provides background information and instructions for performing common maintenance tasks.

Synchronizing Data with Replication

The ADMADE Data Manager is designed to be a multi-user application. This is needed because there are several non-networked computers at the ADMADE headquarters in Nyamaluma, and the application may also used on laptops in the field or at offices in Chilanga. Despite their different locations, all users need to be able to have access to the most recent data, as well as a means to share new data entered with other users.

The ADMADE Data Manager solves these multi-user needs by making use of Access's built-in ability for database replication. Replication, which is described thoroughly on the Access online help system, basically involves creating a Design Master of the database and one or more Replicas. The Replicas are used primarily for data entry and querying, while development on the Design Master can include creating new forms, reports, tables, etc.

A key element in maintaining a replicated database system is synchronization. When two databases, such as the Design Master and one of the replicas, are synchronized, new data is shared between the two and new objects from the Design Master are copied to the replicas. In practice, synchronizing requires copying the main MDB file from one computer to the other using a LapLink cable and Direct Cable Connection (or similar software), using the Synchronize command on the Access Tools menu, and then copying the file back to the original computer.

The danger of using replicated database (and any other multi-user database for that matter) is that accidental data changes are broadcast to all replicas during synchronization. Hence, if a novice data entry clerk accidentally deletes or edits records from a table, the next time the databases are synchronized those records will also be deleted or altered in the other replicas.

To prevent errors from rippling throughout the database, you can use Partial Replicas or non-synchronizable copies of the database for those users who don't need the most recent data. You can also make it a practice to check the Data Log before synchronization to see what records have been altered. But the best prevention is to ensure that people entering data are properly trained and supervised.

It is important that replicas get synchronized frequently, especially during periods of data entry. If too many records have been added in each file since the last synchronization, then Access may encounter conflicting records and synchronization errors can result. Synchronizing also reduces the size of the database, which can save a significant amount of hard disk space.

Partial Replication

Partial replicas contain the same queries, forms, reports, and modules as any other replica, but differ in that they only contain a sub-set of the records of a full replica. The advantages of using partial replicas include reduced file size, and you can restrict which records users have access to (which can help prevent accidental erases or edits).

Partial replicas might be a good idea when you don't need users to see all records in all tables. For example, if you need to make a replica that will only be used for data entry, the replica doesn't need to contain many of the older records. This can make the size of the replica smaller and also prevent the data entry clerk from accidentally tampering with older records.

Other notes about partial replicas:

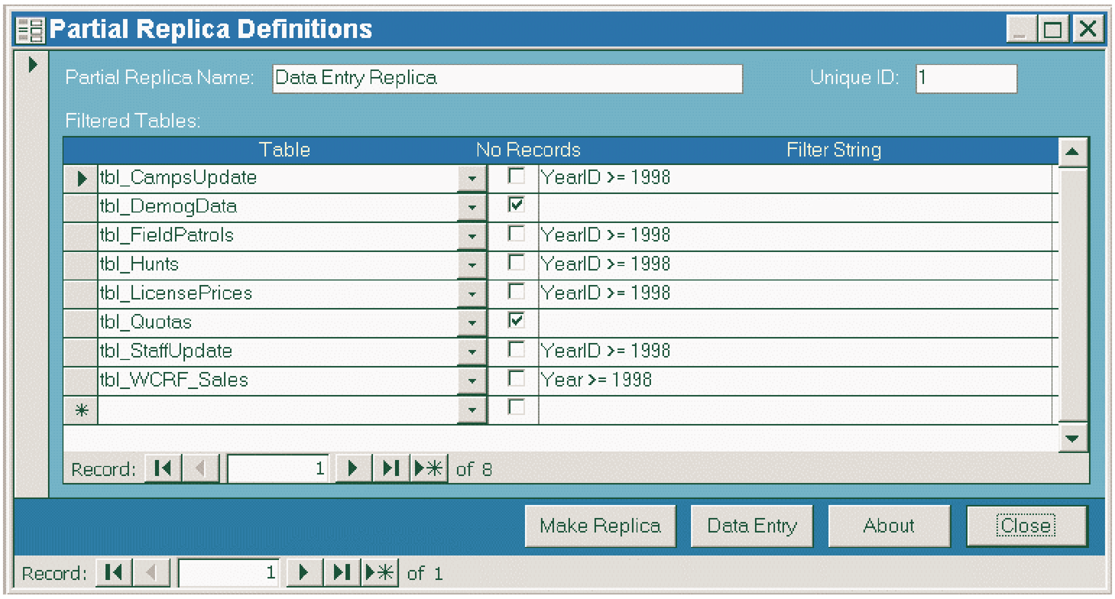
- Unlike full replicas, you can't make partial replicas using commands from the Tools menu. You can only make partial replicas using the Partial Replicas Definitions form (see below) or another programmatic method.

- You can still synchronize partial replicas the same way you would synchronize full replicas. However you can only synchronize partial replicas with a full replica (or the design master). You can't synchronize a partial replica with another partial replica.

- When you synchronize a partial replica, only those new records from the design master that meet the criteria in the partial replica will be added to the partial replica.

- To create a partial replica, you need to specify a filter for those tables that you wish to have reduced records. Each table must have its own filter expression, you can't use one filter for the entire database or multiple tables. Although you can create a filter for any table, normally you just need to create a filter for those table which contain a lot of data, such as field patrols, safari hunts, demographic data, etc.

To create partial replicas of ADM, open any full replica and select Partial Replicas from the Main Menu.



Partial Replica Name: Data Entry Replica

Unique ID: 1

Filtered Tables:

Table	No Records	Filter String
tbl_CampsUpdate	☐	YearID >= 1998
tbl_DemogData	☑	
tbl_FieldPatrols	☐	YearID >= 1998
tbl_Hunts	☐	YearID >= 1998
tbl_LicensePrices	☐	YearID >= 1998
tbl_Quotas	☑	
tbl_StaffUpdate	☐	YearID >= 1998
tbl_WCRF_Sales	☐	Year >= 1998
	☐	

Record: 1 of 8

Record: 1 of 1

Each Partial Replica definition should have a unique name and ID number. In the Filtered Tables sub-form, list those tables that you want to be filtered in the partial replica. If you want no records from a table to appear, check No Records. Otherwise enter a filter string for one of the fields in the table.

In general, you don't want to put filters on lookup tables, because referential integrity requires that these records be kept because there are related records in other tables. If you set a filter expression on a lookup table but another table has related records than the records will remain in the table despite your filter.

When you are done entering filter expressions for all of the tables you want to be reduced, click the Make Replica button to make the partial replica.

Compacting and Repairing ADM

Like any Access 97 database, ADM needs to be repaired and compacted from time to time. Repairing an MDB file fixes a variety of mostly minor problems that accumulate when a database is shut down prematurely or even used normally. These minor errors include invalid entries in table indices and bad records in the system tables.

Compacting a database recovers the space from deleted records and objects, reducing the size of the MDB file on the hard disk. Compacting a database is particularly important in replicated databases, which can become *quite* large due to the extra information they have to store. Compacting should also be done before two databases are synchronized.

Unfortunately Access is not smart enough to tell you when your database needs to be repaired or compacted. In general it is up to the user to know when the file needs compacting or repairing, but to help remind you when compacting is needed, the Main Menu has a feature to automatically detect when a file is becoming too big. It may also be a good idea to use the Menu System's Reminder Messages feature to remind yourself to repair and compact a database once a week or so.

You can repair and compact database files by selecting *Repair* or *Compact* from *Database Utilities* on the Tools menu.

Backing Up ADM

In traditional data management, a backup is usually a copy of a file saved at different locations at specified intervals. However with a replicated database, backing up is different both conceptually and in practice. In essence, each replica of ADM can be considered a backup, because it contains all the objects in the original. Furthermore, each time replicas are synchronized, data sets are updated and design changes are propagated to other members of the replica set. This is in essence also a backing up process. Thus, as long as replicas are synchronized on a fairly regular basis, copies of the data are being made and additional backups are not required.

Although it is true that design changes can only be made to the design master, any of the replicas can be converted into the design master at any time if something should happen to the original design master. This should be avoided of course unless the original design master gets lost or corrupted.

You can and should, of course, also make 'traditional' backups of ADM from time to time by making a duplicate copy of one of the replicas on a zip disk or floppies. This is particularly important because errors can be broadcast throughout the database during synchronization without being apparent.

Restoring a backup and using it as a member of the replicas set should be discouraged, however, because when you restore this back up and attempt to synchronize it with other replicas in the set, you may get synchronization errors. If the design master has been lost, it'd be better to convert one of the replicas into the Design Master. If records have been accidentally deleted or altered, it'd be better to restore those records from the backup by creating links to tables in the restored database and using append or update queries. See the Access online help system and the chapter on replication in the Building Applications manual (on the Office-97 CD) for more information about replicas.

File Layout

In addition to the Microsoft Access files, some forms in the ADMADE Data Manager also depend on GIS coverages (e.g., ESRI shape files) and/or image files, and/or documents. The standard location for these files is illustrated in the following directory tree:

```
C:\ADM\
    |  ADM_xxxxxx.MDB            Main Access replica file
    |
    +--Covers\
    |         (all shape files)
    |
    +--Images\
    |         (all image files – pictures of animals)
    |
    +--Docs\
              (document files)
```

Notes:

- the files listed above may not be the **only** files needed to run ADMADE Data Manager. In addition to all the MS Access files needed, you may need to install Map Objects. See Installing ADM on a New Machine.
- in general you should NOT rename or move these files unless necessary because you might disrupt the linkages between them.

Installing ADM on a New Machine

Because the ADMADE Data Manager uses MS Access, which is an off-the-shelf database package, it is fairly easy to set up on a new computer. It can be installed on a stand-alone machine or put on a network for multi-user access.

Installation Requirements

- You must have MS Access 97 installed on the destination computer.
- In order to see the interactive charts and graphs, MS Graph should be installed (MS Graph is automatically installed when you do a standard installation of MS Access. However it may not have been added if you did a compact or custom installation)
- PowerPoint must be installed in order to use Interactive Maps and Slide Templates.
- For best results you should have 32M of RAM or more
- You will need about 50Mb of hard disk space for a full replica and all the external files (e.g., covers, images, documents). If you don't have this much, you can omit the external files and/or use a Partial Replica.

Standalone Installation

1. Copy the contents of C:\ADM and all its sub-directories to the new machine, *except* for the main ADM_xxxxxxxx.MDB file.

   Note: if possible, recreate the same directory structure on drive **C** on the destination machine.

However if for some reason you need to put the files on drive D or E, then the first time you start ADM_xxxxxxxxx.MDB you should go to Admin => Program Options and enter the correct path to the GIS covers and image files,

2. Make a new Replica from the Design Master or another full replica in the set using the Create Replica command on the Tools – Replication menu. Give this file a new name, such as ADM_REPLICA4.MDB, and copy it to the new computer.

3. Give the BMP file in C:\ADM directory the same name as the main MDB file (except of course for the BMP extension) so that the splash image will appear when the database is first opened.

4. If not already on your system, the Map Objects Active X control needs to be installed separately using the special ADM Map Objects setup program. You may also need to install DAO3.0 if your machine doesn't already have it. Use of this control requires the appropriate license.

5. Open the ADM_AddButtons.mdb file to run a program which will add buttons to your Window menu and toolbars. These buttons are not required but can be used to open the Main Menu, Filter Manager, Reference Map, Caption, and About windows.

Installation on a Peer-to-Peer Network

Because ADM uses database replication instead of a front-end/back-end structure for multi-user support, there are no special requirements for installing it on a peer-to-peer network. For maximum performance, each workstation should have its own replica. Alternately, multiple workstations can open the same replica file from the server. When using replicas, regardless of whether they are on a network or not, remember to synchronize frequently to prevent synchronization errors and divergent datasets.

Glossary and Acronyms

ActiveX – a type of software extension that you can use in programs such as MS Access to do things that you couldn't normally do. For example, with a MapObjects ActiveX control you can view GIS covers on your forms.

CBD – Community Based Distributor – a resident of an ADMADE GMAs who is responsible for distributing family planning resources. Some GMAs have CBDs.

classical safari – a safari hunt which lasts for 7 or more days where the hunter can shoot up to 14 animals including the big cats

classified – coloring features on a map based on numeric values

client – an international hunter who comes to Zambia

concession fee – a flat fee that a safari hunter must pay for hunting in a Game Management Area. There are different concession fees depending on whether the hunt is a classical safari or mini-safari

design master – in a replicated Access database, the design master is the replica where you can add or make changes to the design of replicated objects. See the MS Access help for more info

ESRI – Environmental Systems Research Institute. The company that makes ArcView and ArcInfo GIS software

GIS – Geographic Information System. A combination of software, hardware, and data you can use to work with spatial data. You can make maps with GIS software

GMA – Game Management Area. A semi-protected area in Zambia. There are more than 30 GMAs in Zambia, many of them buffering national parks

hunting block – an area which is designated for safari hunting. Usually one GMA will have one hunting block, but sometimes there is more than one hunting block in a GMA, and still other hunting blocks are spread across more than one GMA

license – a per-animal fee which is charged to safari hunters. License fees vary according to the species

mini safari – a hunt which can not exceed 7 days, can not shoot more than xx animals, and can not take certain species including roan, eland, lion, and leopard.

NPWS – National Parks and Wildlife Service. The government department under the Zambia Ministry of Tourism responsible for all wildlife management in the national parks and game management areas. NPWS was transformed into the semi-private Zambia Wildlife Authority in 1999.

ODBC – Open Database Connectivity. A set of software drivers that you can use to retrieve data from a variety of databases. For example, you can use ODCB in ArcView to link to data saved in a MS Access database operator – a safari company (usually privately owned) that has the concession lease for a game management area. The safari operator usually contracts out to one or more Professional Hunters to oversee running the safari camp in the area and guide clients

professional hunter – a hunter who is contracted by a safari operator to run a safari camp for visiting international hunters. Being a professional hunters is typically a seasonal job, and they often live in urban areas the rest of the year

PKZIP – a format for files which have been compressed, meaning that they take up less space on the hard

replica – in MS Access, a replicated database is a database that has been divided into duplicate files, or replicas. Each replica can be used on a different computer, and you can synchronize the replicas so that new data is shared between the replicas

RDMS – Relational Database Management System. Any database system that is relational, meaning that data is stored into different tables that are related together by common key fields

referential integrity – a feature of MS Access whereby the integrity of the data in related tables is automatically enforced. For example, when a relationship between a lookup table and details table is defined and enforces referential integrity, then no record can exist in the details table (e.g., record for a field patrol) without a valid corresponding record in the lookup table (e.g., a Unit). Likewise, when referential integrity is enforced, you can't delete records from a parent table if it will leave orphaned records in a details table.

SQL – Structured Query Language. A universal language understood by many databases to extract information from database files. Example of a simple SQL statement: SELECT FirstName, LastName FROM tbl_Employees WHERE FirstName='Ngulube';

Shape File – one of many file formats for GIS data. Shape file is the native format for ESRI products such as ArcView and MapObjects

Unit – an administrative body in ADMADE which manages the program in the field. A Unit often corresponds to a single Game Management Area, and will typically be managed by a Unit Leader, support staff, scouts, and community level organizations

Universal Grid Number – a six digit number which uniquely identifies a grid in ADMADE. The first two digits correspond to the ID number of a Unit or Hunting Block. The last four digits represent the grid number which is printed on the base maps. Universal Grid Numbers are required when displaying grid-based data in maps.

VAG – Village Area Group. An administrative sub-division of an ADMADE Unit, representing several neighboring villages. A Unit will typically consist of 3-6 VAGs

WCRF – Wildlife Conservation Revolving Fund. A fund which was set up by NPWS around 1991 to administer revenue from safari hunting and other income from wildlife use. A portion of the revenue to the WCRF goes to communities for resource management and community development

ZWA – Zambia Wildlife Authority. The semi-private institution that evolved from/replaced the National Parks and Wildlife Services in 1999